esprit

Se former autrement

2018 - 2019

GRADUATION PROJECT

NATIONAL ENGINEERING DEGREE

SPECIALTY : INFORMATION TECHNOLOGY (SIM)

Development of GuestWiser

By: *Beyram Ben Elghali*

Academic supervisor: *Imed Amri*

Corporate Internship Supervisor: *Giovanni Moretto*

METIDE

cdio

EUR-ACE

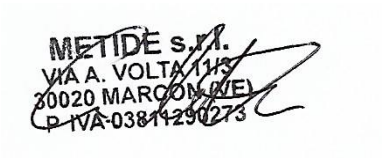
Cti

CONFÉRENCE DES GRANDES ÉCOLES

Signature

Academic Supervisor signature

Corporate Internship Supervisor signature



Dedications

I dedicate this modest work to my dear parents **Atef** and **Nesrine**, my greatest treasure, for their encouragement and support throughout my studies. May God keep you and bless you for all the good and love you have given me.

To my brother **Badis** to whom I hope a life full of joy and happiness and success in his studies.

They gave me all the beautiful moments we had together and supported me at the worst of times.

To the whole **Ben Elghali** family for their encouragement and dedication.

To all **teachers** from primary to university for their hard work that led to my success.

To all my **friends** and all those who are dear to me.

To all those who helped me, from near and far.

Thank you all from the bottom of my heart.

Acknowledgments

It is with great pleasure that I reserve this page in deep gratitude to those who supported and encouraged me throughout my graduation project and guided me all the way to make a better work.

My thanks go to my Internship supervisors Mr. Giovanni Moretto and Mr. Andrea Maglie for giving me the chance to learn more and for sharing your knowledge with me which was definitely helpful for the realization of the project.

My thanks also goes to Mrs. Imed Amri, my supervisor at ESPRIT for his availability, his directives and his very enriching advice and her fruitful help.

I would also like to thank especially the members of the jury who accepted to judge this modest work.

Finally, I would like to thank all those who contributed to the accomplishment of this project under the best conditions.

Table of Contents

Table of Figures

Table of Tables

Introduction

Over the past few years, technology has evolved at an interesting stride, it has revolutionized our world. This great evolution has greatly served humanity in its environment and also to considerably improve its standard of living by creating amazing tools and resources, putting useful information at our fingertips in order to facilitate humanity's tasks. So it becomes possible to share the information and remote access to everything you need.

All technologies are born out of purpose. When it comes to lodging, booking and rental management, modern technology has had an impressive influence and could solve almost every problem and every issue related to this domain. Several companies are exploiting this advanced technologies to offer innovative and fast services to their consumers.

GuestWiser is a technology service provider based in Los Angeles that represents a property management system which gives the possibility for guests to arrange theirs reservations among GuestWiser apartments and managing their facilities like Parking, Wi-Fi, Building Access etc. in order to bring conformability, security for their guests and to remove the boundaries and limitations of the short-term stay.

My mission was to create this system which is based on developing the back-office web application that can show all the data for the reservations, listings, chat service for the administration, and both Android/ iOS mobile applications for the guests to give them the possibility to manage everything about their bookings.

The report contains six chapters, in each chapter we analyze the different steps of designing and implementing this project, it goes through a whole well-organized structure for the built system as described below:

- State of art: In this chapter we will present the company where the work was accomplished, we give a detailed description of the project, introduce the existing solutions and discuss their functionalities from a critical point of view, present the proposed solution, and explain the work methodology.
- Sprint 0 | Requirements Analysis: In this section we will distinguish the actors, explain the project's functional and non-functional and sort all the features by priority for each module.

- Solution Implementation: this section contains a specific study about the realization of our solution which is about the working environment, the architectural pattern and the deployment, package diagram that can explain the general behavior of the application etc.
- Sprint 1 | Web Administration Portal: this chapter will explain the flow for implementing the web application and present the use cases and the product backlog.
- Sprint 2 | iOS Mobile Application: this chapter will explain the main steps for the development of the iOS mobile application by providing uses case and sequence diagrams side by side with a sprint backlog.
- Sprint 3 | Android Mobile Application: this chapter will explain the main steps for the development of the second mobile application on Android platform by providing uses case and sequence diagrams side by side with a sprint backlog.

The final section of this report will include a general conclusion, as well as the deduction of possible improvements to our systems.

CHAPTER 1: State of art

Introduction

State of the art is the high-level and the advanced step of every single research and development Technology. In this chapter, we will introduce the general context of the internship's project. First, we are going to introduce the host company and its different working fields. Then, we are going to make our preliminary study starting with the existing solutions in order to understand the goals behind the selected subject. Finally we will present ours as well as the team working methodologies.

Presentation of the host company

About the company

Founded in 2008, Metide is a mobile agency located in Venice, Italy, specialized in the design and development of applications for both mobile and web platforms. The company mission is to guarantee to the customer advanced and high performance solutions, studied and designed according to its needs.

Fields of work

Figure 1 Metide logo

The company working process favors corporations that consider the development of products for both B2B and B2C markets in order to improve the internal processes and to facilitate interaction with end customers Metide put its expertise at the service of companies, placing itself as a trusted consultant alongside them. Since it was founded, Metide has closed more than 70 projects in different technologies such as Mobile applications in both native and hybrid development or web applications.

Company structure

The following image represents the Metide team's structure and the leading members that were actively present during my project's realization:

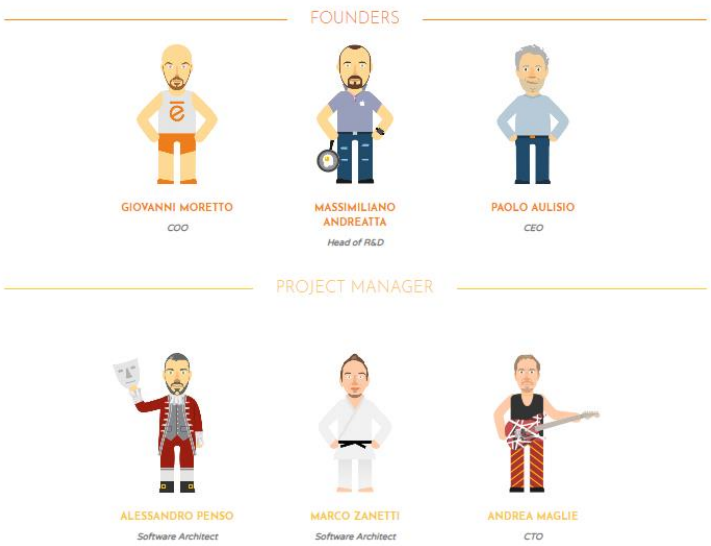


Figure 2 Metide company structure

Project presentation

After introducing the company's goals and activities, we will describe this project and all its modules by providing, the project scope, the problematic and then the existing study including the strengths and weaknesses of each solution, in order to propose our own.

Project scope

This project was conducted during the last year of my studies at ESPRIT in order for me to get the national engineering degree in computer science. It was a very exciting experience to spend 6 Months in Venice, Italy and being able to participate in many international events like DroidCon 2019 in Turin which was one of the best android events in the world and meeting Google Experts from all over the world.

During the period of the internship, this project was supervised by two main coaches in Metide who are Andrea Maglie (CTO) and Giovanni Moretto (COO), as well as an academic coach in the person of Mr. Imed Amri.

Problematic

Accommodation, Short term lodging is one of the most trending subject nowadays in United States of America, but the main problem is that there is too many companies are trying to solve and create solutions for this kind of issues.

The following charts based on financial data in United States show how the lodging industry chains fit into the overall global accommodations picture and how it will evolve by the end of this decade. These projections also take into account recent merger announcements.

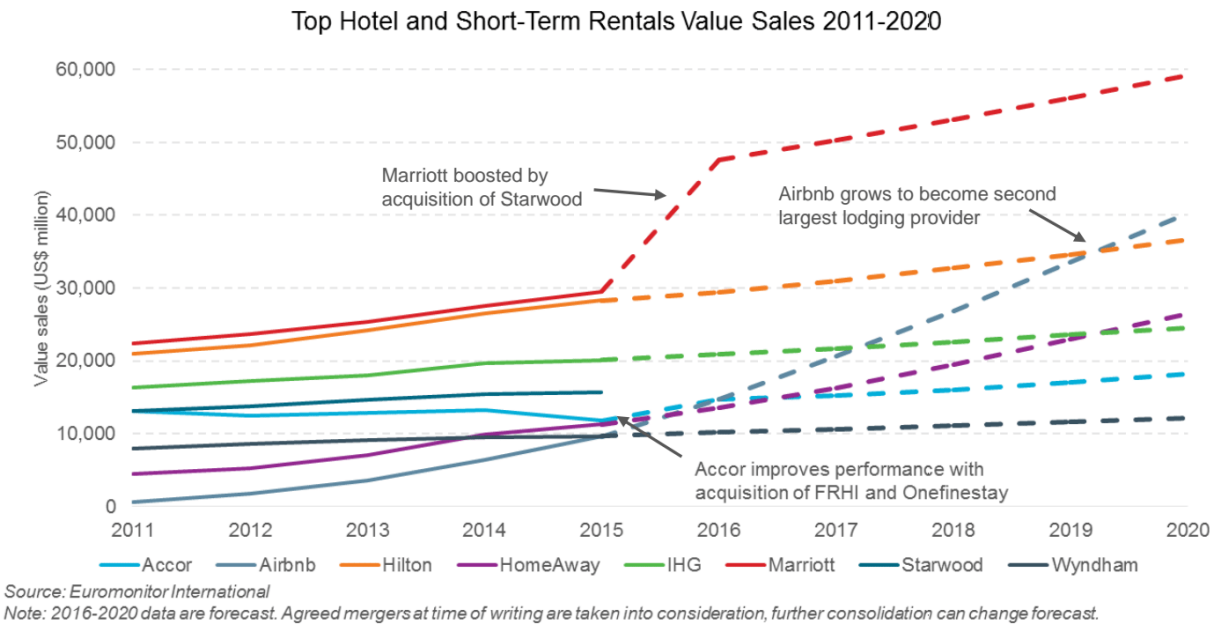


Figure 3 Top short-term rentals values sales

Moreover, gathering client's confidence is a very hard process since searching for the best apartment's deals is not that easy to find and companies tend to make lots of campaigns to build a strong brand image of their company which makes this domain very hard and tricky after the appearance of biggest accommodation worldwide company Airbnb.

Furthermore, there is no reliable guaranty at least for getting all the common apartments amenities already published or about the legal situation for the rental after that The New York state attorney general's office released a report on Airbnb listings that revealed this unfortunate fact and local laws in New York make it illegal.

On the other hand many rental companies like Airbnb don't offer their benefits in the right way like reception, housekeeping or room service. An important thing is that Privacy is never guaranteed and guest are always at the complete mercy of their hosts even for their Lack of safety and security regulations in the apartments

Finally, concluding why you should reconsider services like Airbnb even if there are certainly benefits of using them but only if they are done in an ethical and legal manner.

Existing solutions

In this section, we will handle the market and the existing study. Firstly, we will list all available and public solutions that can be used in managing accommodations by providing a detailed listing of strengths and weaknesses for each one of these systems. Thirdly and finally we will present our solution that can solve these problems.

1. Airbnb

Figure 4 Airbnb logo

Founded in August 2008, Airbnb is an online marketplace available as a web applications as well as Android/iOS application that connects people who want to rent out their homes with people who are looking for accommodations in that locale. It's. It currently covers more than 81,000 cities and 191 countries worldwide.

a. Strengths
   - Cost Savings: Airbnb's prices are generally stable, whether you book last-minute or well in advance, you'll pay the same rate. On the other hand, hotel room rates go up or down based on the day of the week and the time of year.
   - Filtering Tools: Users can look for desired accommodation by filtering their travel like Dates, room type, price range, required amenities etc.
   - Maps: Airbnb simplifies the reservation process using maps. They help in showcasing the location with rentals, around the area the user wishes to stay in.

b. Weaknesses
   - It Isn't Legal Everywhere: The Company violating the Housing Laws and Regulations in many countries. As per many Housing Laws, hoteling and guest houses are prohibited in residential areas.
   - What You See May Not Be What You Get: The Airbnb is dependent on quality provided by the hosts. It the hosts do not follow standard guidelines it can damage the company reputation.
   - Taxes and fees: Airbnb imposes a number of additional taxes and fees, guests could pay a guest service fee up to 20% on top of the reservation fee, to cover Airbnb's customer support and other services.

2. Booking.com

Figure 5 Booking.com logo

Founded in 1996, Booking.com is an Online Travel Agency based in Amsterdam presented as a website and also as a mobile application for Android/iOS. It's a platform that show case and connects travel services like hotels, flights, trains etc. with travelers.

a. Strengths
   - Clear pricing with no hidden fees: The prices displayed are the prices you pay. Booking.com doesn't add any reservation or admin fees.

b. Weaknesses
   - Same as Airbnb, Booking.com can't guarantee amenities of each accommodation.

Proposed solution

The problems that we are trying to solve are how we can guarantee a guest a better experience in short rental lodging by providing him an authentic and affordable full house experience in a legal way. All these mentioned solutions treat all the presented issues as if they were unrelated, but the truth and the feedback of too much guests confirm the opposite at least for United States.

Based on that, GuestWiser which is one of Metide clients is a well-funded start-up and a strong new-comer in the US apartment rental business, with plans to expand world-wide within the next years are trying to solve these needs. Basically, the solution is creating a new platform that will solve all the previous guest problems by offering them an accommodation consist of all the required amenities like Security, Wi-Fi, Parking and Thermostat etc. not only this but also managing them too in a legal situation. Also, give them the ability to check their reservations and keep them updated with the last best deals. And finally, providing a real time customer care chat in case there is any issues or questions about accommodations in order to ensure a great experience for guests.

The solution will be available as a mobile application available on both platform Android and iOS for guests and also a web portal to manage administration stuff.

Mobile Application (Android/iOS)

GuestWiser will provide a user friendly and interactive interface that enables the user to check, filter his reservations, also it will give the user the possibility to manage all the required amenities like Security, Wi-Fi, Parking and Thermostat as it was mentioned before. Moreover, the user could check all the online deals available in GuestWiser apartments and make online reservations.

Web Portal

This web application will be available only for administration and will represents the different administrations tasks so they can check all active reservation, apartments as well as answering guest's question with the customer care chat widget.

Work methodology

A software development method defines the process of dividing the software development process into separated phases so we can improve the design, product and the project management. Also known as software development life cycle. In this project we are going to work as an Agile Scrum team.

Agile Methodology

Agile is an iterative approach in which customer's needs define the requirements and solutions, it allows to manage projects and helps deal with changes regarding a certain project process based on adaptive planning work iterations, early delivery in order to continue the improvement of the final product. It focuses on compete customer involvement throughout the development cycle. It works great when we are working on solutions from the beginning which is our case.

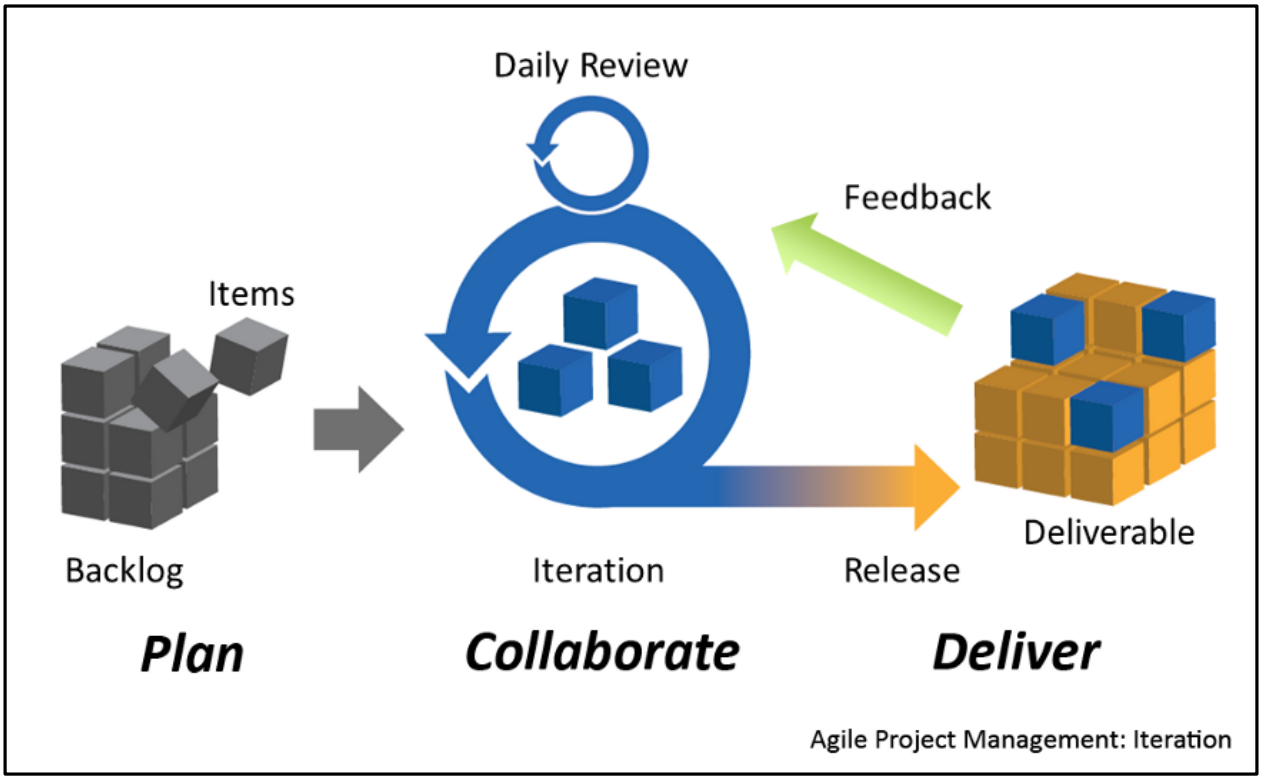


Figure 6 Agile project management iteration

Agile is not a development methodology, it is an approach, a specification that other methodologies follow like Scrum which focuses on task management in a team-based development environment. Scrum works with 5 to 9 members teams, it empowers them to develop a product with mutual cooperation.

Scrum Overview

Scrum brings all the team members to agree on features, after that they need to determine specific goals each sprint which is around 4 week period. At the beginning of each sprint, a sprint backlog should be done, based on the whole product backlog, containing all tasks, priorities, estimations, and assignments. Every day, a standup meetings are organized to consult the progress. The Sprint review and retrospective occurs at the end of the sprint.

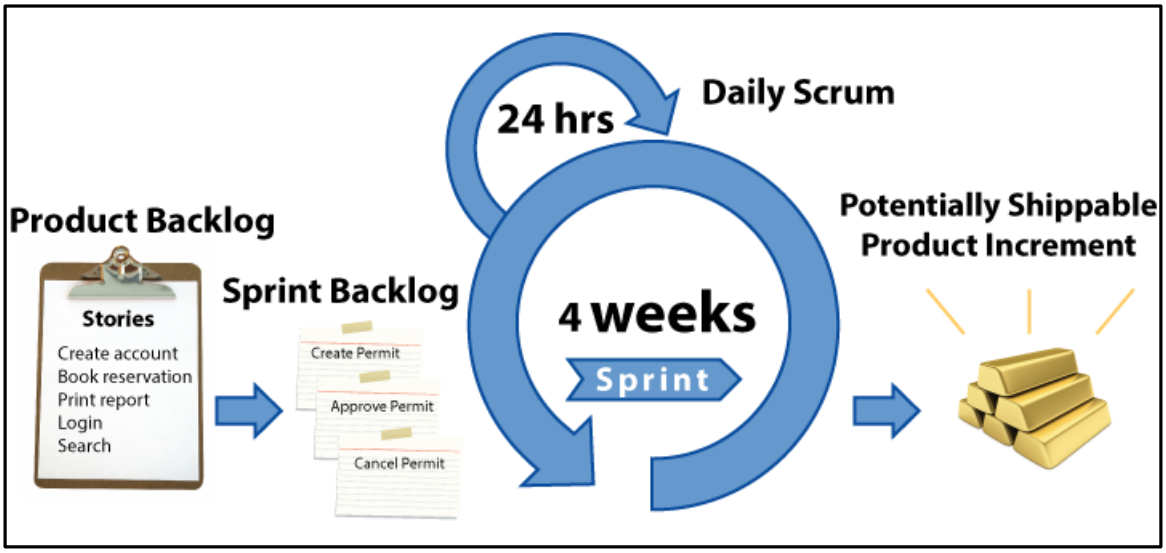


Figure 7 SCRUM iteration

Scrum team

Since we are starting a new Scrum Project, the whole Metide team is involved, so the members are Giovanni Moretto as product owner, Andrea Maglie as Scrum master and myself Beyram Ben Elghali as developer.

- **Giovanni Moretto** is the product owner, he represents the customer best interests and acts more like a business consultant/advisor/coach etc. for the client.
- **Andrea Maglie** is the scrum master, he is responsible of managing the scrum flow, supervising tasks development and he is in touch with the product owner in order to discuss new features.
- **Beyram Ben Elghali** myself, I will be in charge of the development of the Android and the iOS mobile applications for the guests and also the web portal for the administration.

Conclusion

In this initial chapter we had the opportunity to introduce the company and its activities. Then we analyzed the targeted domain by presenting the existing solutions and their issues in order to propose ours. Finally, we studied the working methodology and chose to work with scrum which will guide us to bring the project to the real life.

CHAPTER 2: Sprint 0 “Requirements Analysis”

Introduction

This chapter will contain a detailed study of all project requirements. Firstly, we will distinguish the actors, the functional and the non-functional needs of the project. Secondly, we will sort all the features by priority and make an estimation for each one. Finally, we will present the scrum tools we will be using, a product backlog and sprints planning for the project. This step is known as sprint 0 which is the first step of every scrum-based project.

Identifying Actors

I this part we will identify the different users that will use our application, in this project we have two main actors who are:

• GuestWiser Admin: An admin is a user who can visit the administration web portal in order to supervise all the daily bookings made with GuestWiser he is the business owner, also he can check all the available listings and has the ability to use the customer care chat widget so he can answer the guest's messages.

• Guest: A guest is a normal user that can use the mobile application, he has no administration roles in the application. He can manage his own reservations, and by that control all the amenities available in his apartments. Also he can book from the online deals.

Functional requirements

In this section we will analyze the project requirements, both functional and non-functional ones which will present the behavior of the expected application.

- An Admin can :
  - Login into the web portal with email and password.
  - Update his profile's information.
  - Browse all the integrated reservations and their details.
  - Choose which reservations' properties to be displayed on the portal.
  - Filter Reservations with guest's email.
  - Sort Reservations using any shown property like name, and country etc.
  - Export Reservation into tabular data.
  - Consult the listings and their details.

- Choose which listings properties to be displayed on the portal.
- Filter listings with name.
- Update Parking, Wi-Fi and Building listing's documents.
- Sort listings using any shown property like date, id, status etc.
- Export Listings into tabular data.
- Check Customer care chat messages.
- Filter messages by reservation, email and listing.
- Send Messages.
- Enable the admin to log out from the web portal.

- A Guest can :
  - Create an account via the Android / iOS mobile application.
  - Login into GuestWiser with email and password.
  - Consult all his reservations.
  - Filter reservations by status : active / inactive
  - Check reservation details.
  - Download Parking, Wi-Fi and Building documents.
  - Open main building door.
  - Open/Close listing's door.
  - Check/Change the thermostat's temperature in the apartment.
  - Use Maps in order to get direction to the apartment.
  - Real-time chat with customer care service.
  - Invite other guests to his apartment.
  - Remove guest's invitations.
  - Check GuestWiser Online deals.
  - Browse all GuestWiser Online deals using maps.
  - Check listing's details.
  - Book an apartment using an online payment service.

Non-Functional requirements

- **User experience**
  - The mobile/web application should offer a clear overview of the different functionalities of the application.

- Ergonomic, all the content and the information displayed are clear and the operations are easy to perform for a non-technical person.

- **Security**
  - An authentication process takes place for all the users that respects all roles and access rights for each one.
  - Every data endpoint must be secure and protect both the user data and the GuestWiser data.
- **Notifications**
  - A guest is notified when the status of his booking is changed.
  - A guest is notified instantaneously when he receive a message from customer care chat.

SCRUM project management

SCRUM tools

Scrum tools are used during the development process so we can schedule meetings, elaborate tasks planning etc.

- **Slack:** is a collaboration hub where you and your team can work together to get things done. From project kickoffs to budget discussions, and to everything in between.
- **Clickup:** it is a productivity platform that provides a fundamentally new way to work. It's more than just task management because it offers notes, reminders, goals, calendar, scheduling, and even an inbox.
- **Visual Studio App Center:** it lets you automate and manage the lifecycle of your iOS, Android apps in order to ship it at higher-quality. By connecting GuestWiser repositories, it start automating builds, testing on real devices in the cloud, distributing apps to beta testers, and monitor real-world usage with crash and analytics data.

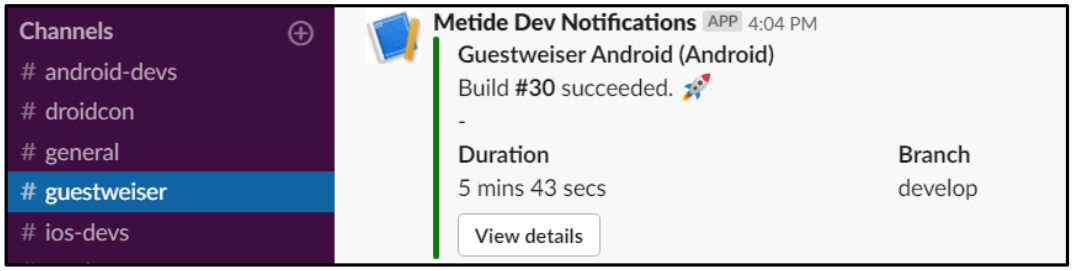


Figure 8 App Center slack integration

Product backlog

It is one of the most important steps in a scrum project which is elaborated during the first Scrum meeting and it contains the needed features, each one contains a number of user stories that are rated by a priority (100 is the highest). Since this project contains two main applications (Android and iOS application will have the same features even if they're developed separately), two backlogs were elaborated.

- **Administration web portal backlog**

id	feature	User story	Priority
WP_1	**Connect to the web portal**	As an admin I want to login into the web portal.	85
		As an admin I want to logout from the web portal.	
		As an admin I want to view all available information.	85
WP_2	**Manage profile's data**	As an admin I want to edit my basic information.	65
		As an admin I want to edit my avatar.	65
		As an admin I want to update my password.	80
WP_3	**Manage reservation**	As an admin I want to browser all reservations.	95

		As an admin I want to check reservation's details.	90
		As an admin I want to be able to filter the reservations with email.	75
		As an admin I want to be able to sort the reservations with a selected property.	75
		As an admin I want to Choose which reservations' properties to be displayed on the list.	70
		As an admin, I want to export the reservations to a .csv file.	80
WP_4	**Manage Listings**	As an admin I want to browser all integrated listings.	95
		As an admin I want to check listing's details.	90
		As an admin I want to be able to sort the listings with a selected property.	75
		As an admin I want to Choose which listing's properties to	70

		be displayed on the list.	
		As an admin I want to be able to filter the listings with name.	75
		As an admin I want to be able to update listing's amenities documents.	95
		As an admin, I want to export the listings to a .csv file.	80
WP_5	**Customer care chat**	As an admin, I want to check all guest's messages.	95
		As an admin, I want to filter all guest's messages by reservation, email and listing.	95
		As an admin, I want send messages to guests.	95

Table 1 Web portal backlog

- **Mobile application backlog**

Id	Feature	User Story	Priority
MA_1	**Registration - SignIn**	As a guest, I want to create an account.	**95**
		As a guest, I want to login into my account.	**95**

		As a guest, I want to view all available information.	**95**
MA_2	**Manage Reservations**	As a guest, I want to consult all my reservations.	**95**
		As a guest, I want to filter my reservations by status.	**80**
		As a guest, I want to check my reservations details.	**90**
		As a guest, I want to download Parking, Wi-Fi and Building documents.	**80**
		As a guest, I want to be able to open the main building door.	**85**
		As a guest, I want to be able to open or close the listing building door.	**85**
		As a guest, I want to check/change the thermostat's temperature in the apartment.	**85**
		As a guest, I want to use Maps in order to get direction to the apartment.	**70**

MA_3	**Customer care chat**	As a guest, I want to send messages to customer care chat service.	**85**
		As a guest, I want to check all the messages from the customer care chat service.	**85**
MA_4	**Manage Invitations**	As a guest, I want to invite other guests to my reservation.	**80**
		As a guest, I want that my reservation should appear for my guests.	**80**
		As a guest, I want to remove guest's invitations.	**80**
MA_5	**Manage online deals**	As a guest, I want to check all up to date deals.	**60**
		As a guest, I want to browse all GuestWiser Online deals using maps.	**60**
		As a guest, I want to check listing's details.	**40**
		As a guest, I want to book an apartment	**40**

		using an online payment service	

Table 2 Mobile application backlog

Global use case diagrams

Use cases are used during the sprint 0 of a scrum project to identify the main functionalities. For our solutions two global use case diagrams are built, the first for the administration web portal application and the second for the mobile application.

- **Web portal global use case diagram**

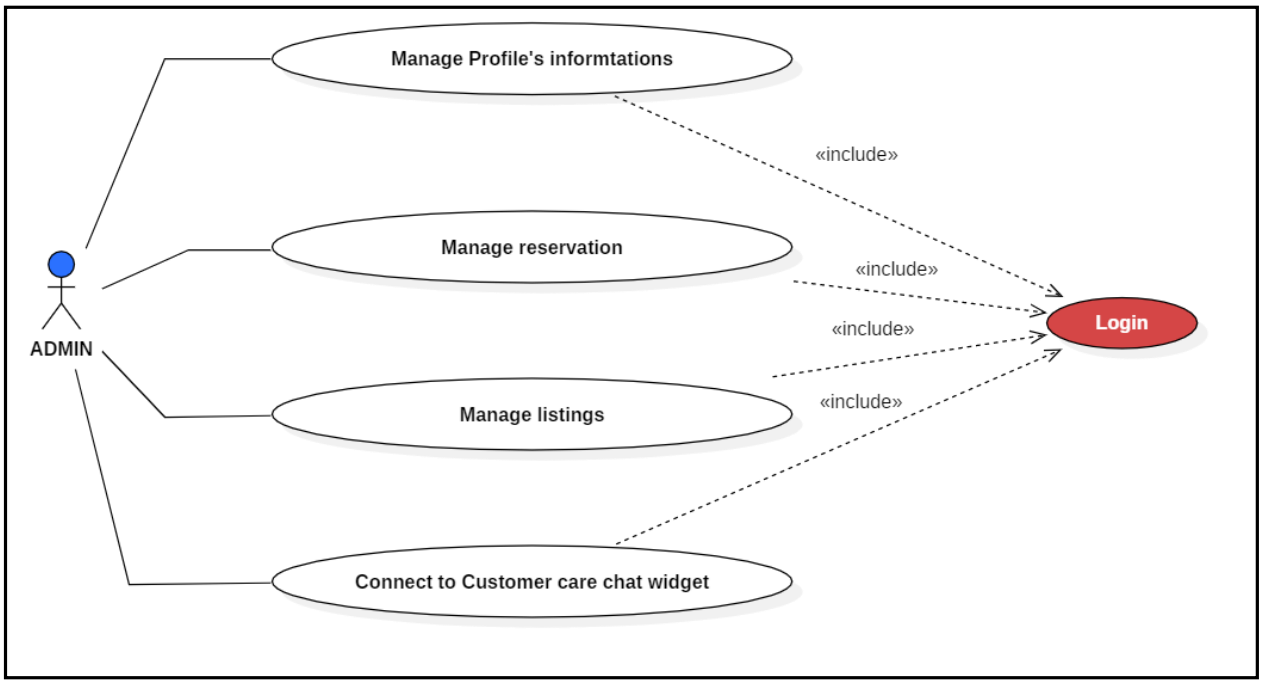


Figure 9 Web portal global use case diagram

- **Mobile application global use case diagram**

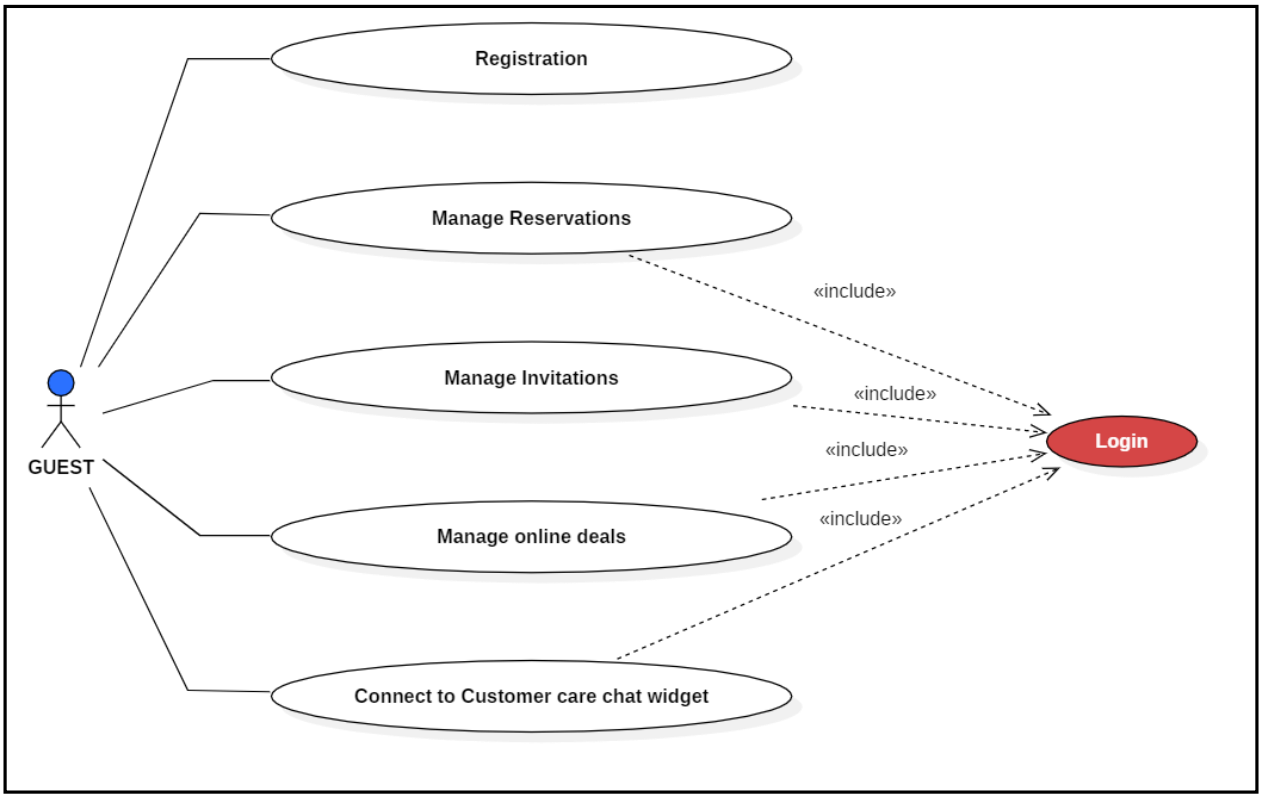


Figure 10 Mobile application global use case diagram

Sprints planning

The last step of the first scrum meeting was separating the project into sprints and giving an estimation of each sprint. This project will be separated into three sprints, this first will include creating the web portal. The second and the third will be dedicated for the development of both Android and iOS mobile application and all related features where their priorities are more than 60.

Sprint 1 : Web Portal	Sprint 2 : iOS	Sprint 3 : Android
WP_1: Connect to the web portal **WP_2:** Manage profile data **WP_3:** Manage reservations **WP_4:** Manage listings	**MA_1:** Registration - Sign In **MA_2:** Manage reservations **MA_3:** Customer care chat service	**MA_1:** Registration - Sign In **MA_2:** Manage reservations **MA_3:** Customer care chat service

WP_5: Customer care chat service.	**MA_4:** Manage Invitations **MA_5:** Manage online deals.	**MA_4:** Manage Invitations **MA_5:** Manage online deals.
From 1st February to 14th of March	From 15th of March to 14th of May	From 15th of May to 14th of July

Table 3 Sprints planning

Conclusion

In this chapter we analyzed all the requirements for this project. We presented the main actors and their needs, then elaborating our product backlog and the global use case diagram and finally we separated the project into three sprints.

CHAPTER 3: Solution implementation

Introduction

In this chapter we will present a study about the realization phase of our solutions starting by the working environment study that will include the hardware and software environments and the used programming languages. Finally, we will introduce the followed architectural pattern and the package and deployment diagrams.

Working Environment

The first section will contains the hardware and software environments, the programming languages and frameworks that were used to develop our solutions.

Environments

Developing our project is based on two different environments.

- Dev Environment: it represents the developer's computers, the machines that were used for writing the source code of our solutions.
- Prod Environment: this is the final production server. Here we will deploy the features that were tested and approved under the Dev environment. It is mainly our final product.

Now, we will provide you with the list of used hardware and software in each of these environment.

Hardware environment

- Dev Environment: we use apple computers with macOS Mojave distribution as an operating system.
- Prod Environment: this environment is running on virtual private servers (VPS) called Amazon EC2 that provides secure, resizable compute capacity. They use Ubuntu 16.04 as an operating system.

Software environment

Android Studio 3.3: Android Studio is the official IDE for Android application development, based on IntelliJ IDEA.

Figure 11 AS logo

Xcode (10.1): it is the development environment for Apple operating systems such as OS X, iOS, WatchOS and tvOS.

Figure 12 Xcode logo

Visual Studio Code (1.31): Created by Microsoft, it is a great code editor which can be integrated with different plugins to improve the use experience.

Figure 13 VS logo

Google Chrome: Created and powered by Google, it's the most used freeware web browser in the world.

Figure 14 Chrome logo

Postman (7.0): Currently it is one of the most popular powerful HTTP client used for testing API and web services.

Figure 15 Postman logo

Apache HTTP: it is the world's most popular HTTP web server creation, deployment and management software.

Figure 16 Apache logo

Programming languages and frameworks

We had to work with several frameworks and languages in order to develop our solutions for various platform. They are separated into two parts. The first one is Back-end and second is the Front-end which presents the mobile applications.

Back-end

Figure 17 Laravel logo

Laravel 5: Laravel is a PHP framework for creating web applications in order to make common development tasks easy such as routing, authentication, sessions, and caching.

Front-end

Figure 18 Android logo

Android: Android is a complete OS. It is not just a framework. The Android software development kit includes everything we need for the development, testing and debugging of android applications using both Java and Kotlin as a programming languages.

Figure 19 iOS logo

IOS: Developers can use the iOS software development kit (SDK) to create applications for Apple mobile devices. The SDK includes tools and interfaces for developing, installing, running and testing apps using Objective C and Swift as a programming languages.

Figure 20 Angular logo

Angular: is a structural framework for dynamic web apps. Created and maintained by the angular team at Google. We used this framework for the development of the customer care chat web widget.

Architectural pattern

An architectural pattern is a general and reusable solution for a commonly occurring issue in software architecture within a given context. In GuestWiser, firstly, we have our main server which will be responsible for managing all the business logic and have an access to the data tier or simply the database server, it will store data and only communicate with the application server.

Secondly, Fantastic Stay server which will be related for the integrations of our listings into other external solutions like Airbnb and Booking etc. also as a first step the reservation flow will be done with Fantastic Stay. Thirdly we have Point Central server which will provide all the amenity services in the apartments like Wi-Fi, Thermostat, doors etc.

Furthermore, we'll use firebase cloud server in order to stores data related to customer care chat service and finally, we have our client represented by the mobile application and the administration web portal. They will request all our services and the main server provides them with the requested data. The clients and the server are communicating using http requests, the next figure shows how the physical architecture of our solution is distributed:

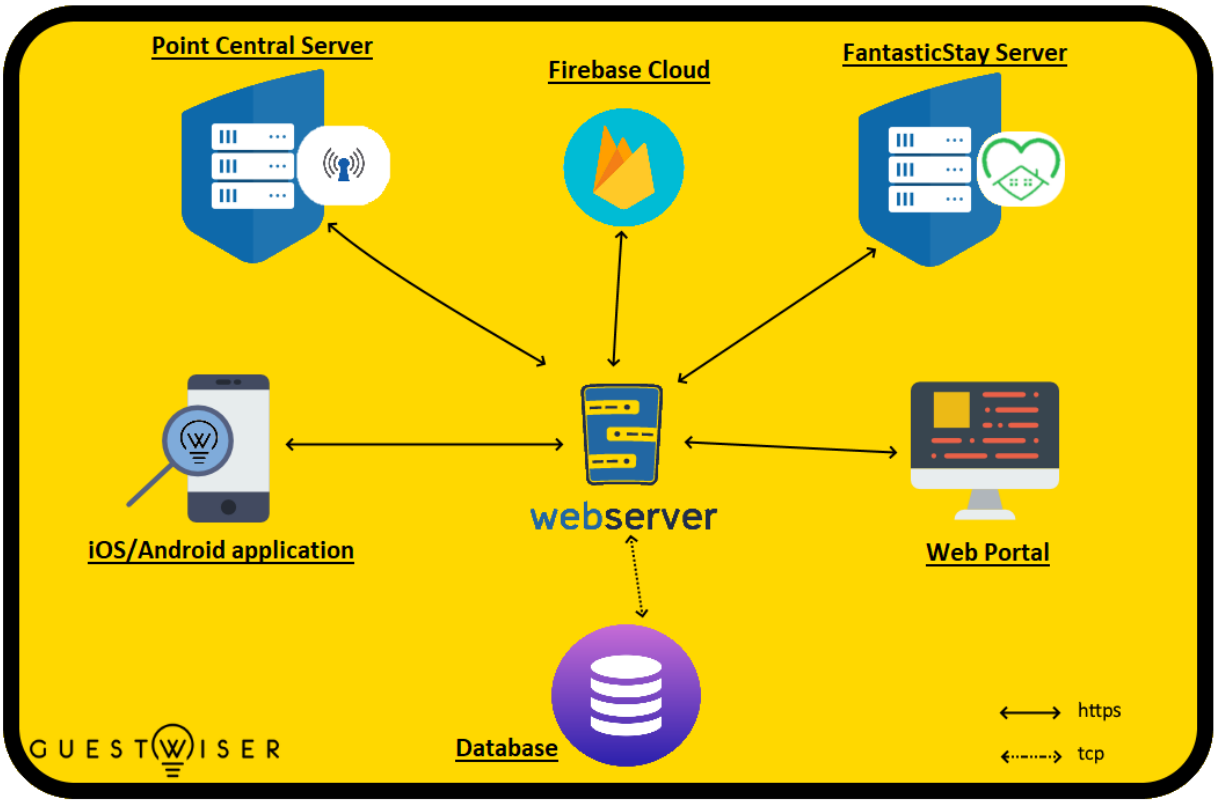


Figure 21 GW architectural pattern

To conclude, we have our database that provides data storage to an application server which itself perform restful web services to our two clients and communicate directly with other external servers in order to do all the business logic in the project.

Package diagram

A package diagram is a structural diagram that follows hierarchal structure of nested packages in order to simplify complex class diagrams. In our project we have four different sub-systems, which are our back-end and Point Central back-end, Fantastic Stay back-end, Firebase Cloud storage, and two different front-end projects which are the web portal and the mobile applications for both iOS and Android platforms. These subsystems connect to each other using Restful web services.

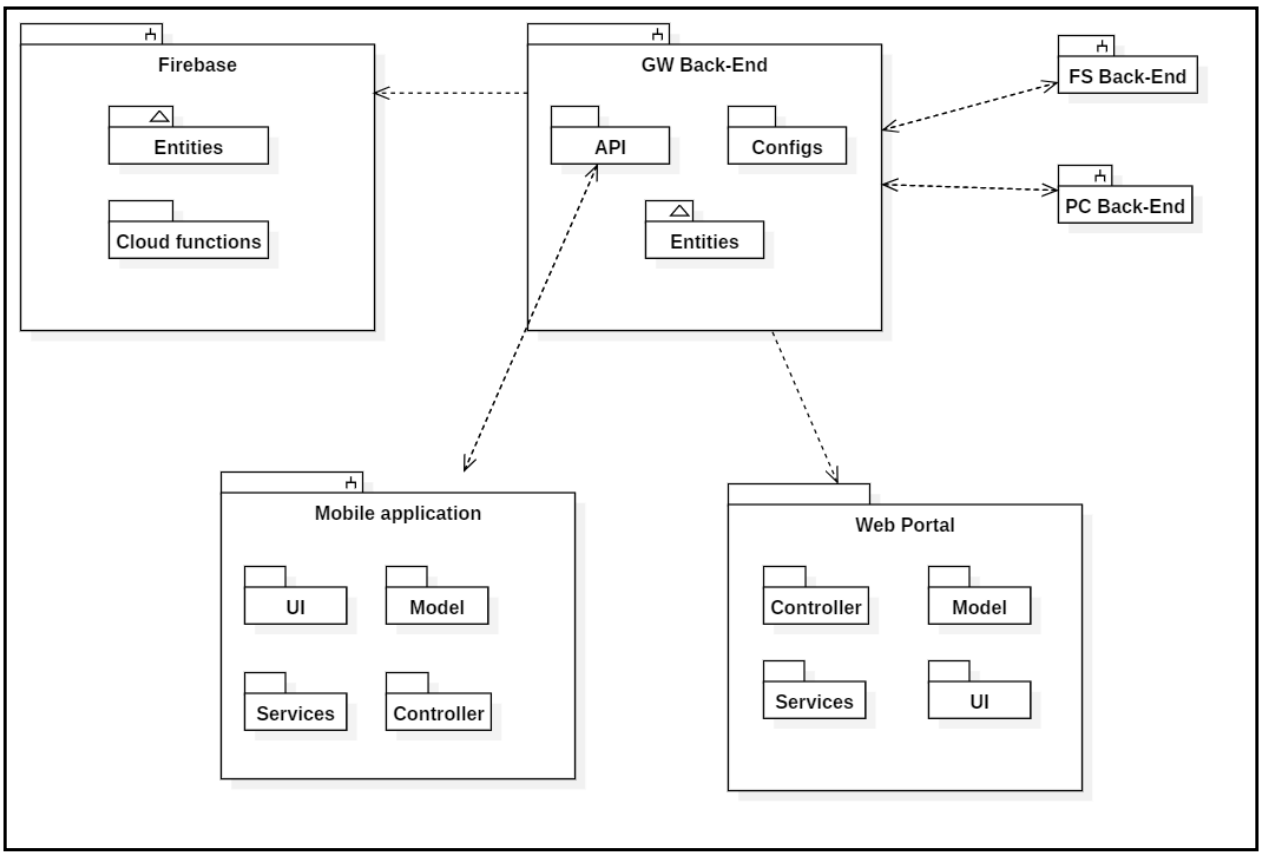


Figure 22 Package diagram

Deployment diagram

The deployment diagram is a kind of structure diagram used in modeling the physical aspects of an object-oriented system, it describes the architecture as concrete nodes along with both internal and external dependencies and connections, known as artifacts. In our case the nodes are the clients, Firebase Cloud server, Fantastic Stay server, Point Central server and the main server which runs two execution environments, the database server and the backend server.

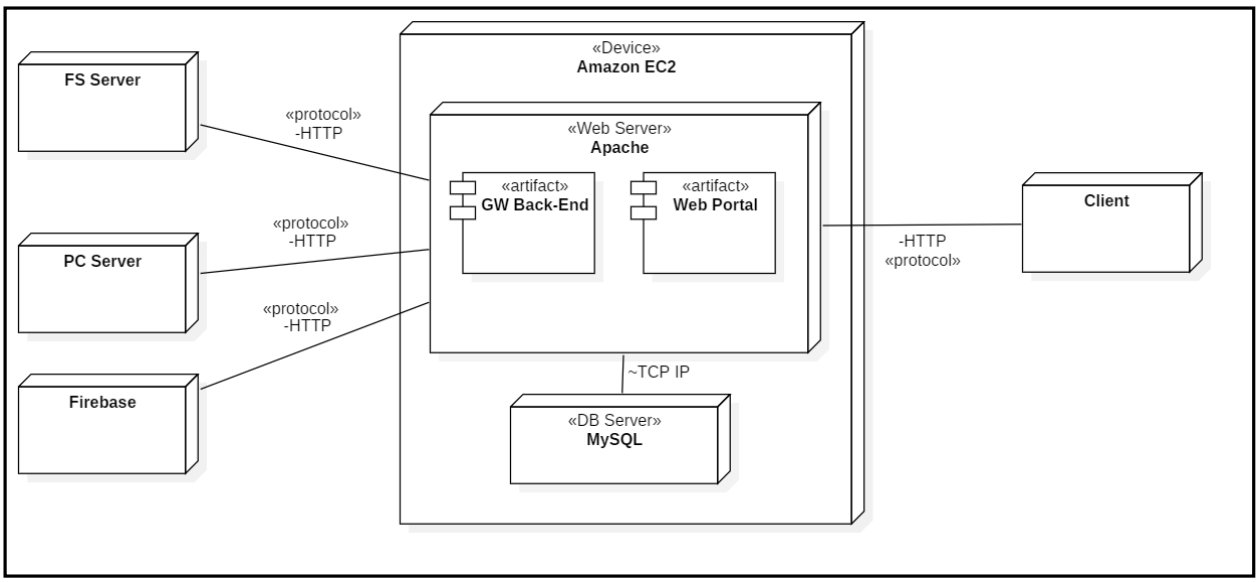


Figure 23 Deployment diagram

Conclusion

In this chapter, we explained and designed everything related to the implementation of our solutions starting with the chosen technologies, the architecture pattern, side by side with the deployment diagram related to our project.

CHAPTER 4: SPRINT 1 “Web Portal”

Introduction

In this chapter, we will present the first sprint of this project which will hold the web portal application for GuestWiser administration and developing the tasks with priorities between 50 and 100 knowing that 100 is the highest. First, we will describe all the user story and its needed tasks. The next step is the conception of the project, it contains the class diagram and a detailed sequence diagram. Then we will begin developing our web portal while keeping track of each task so we can elaborate a burn down chart at the end of this sprint.

Functional specifications

We determined the functional specifications in the first scrum meeting of this first sprint, also, we created a sprint backlog then created a sprint detailed use case diagram.

Sprint Backlog

The first sprint backlog will contains all the tasks that we will be working on, we added an estimation in hours for each task in order to keep track of our timeline. The following table summarizes the product backlog of our first sprint.

ID	User story	Task	Time(h)
WP_1.1	As an admin I want to login into the web portal.	create admin login API	6
		create login UI	3
		test user story	2
WP_1.2	As an admin I want to logout from the web portal.	create admin logout API	6
		integrate logout button in the UI	6
		test user story	2
WP_1.3	As an admin I want to view all available information.	create getDashboard API.	5
		create dashboard UI	2
		test user story	2

WP_2.1	As an admin I want to edit my basic information.	create updateAdminInfo API	5
		create update info UI	3
		test user story	2
WP_2.2	As an admin I want to edit my avatar.	create updateAvater API	6
		integrate picker button in the UI	3
		test user story	2
WP_2.3	As an admin I want to update my password.	create updatePassword API	3
		create update password UI	1
		test user story	2
WP_3.1	As an admin I want to browser all reservations.	create getAllReservations API	10
		create reservations UI	3
		create ReservationFilter API	3
		create ReservationSort API	3
		create exportReservation API	2
		test user story	2
WP_3.2	As an admin I want to check reservation's details.	create getReservationDetails API	4
		create Reservation details UI	2
		test user story	2
WP_4.1		create getAllIntegrations API	9

	As an admin I want to browser all integrated listings.	create Integrations UI	3
		create IntegrationFilter API	5
		create IntegrationSort API	5
		create exportIntegration API	2
		test user story	2
WP_4.2	As an admin I want to check listing's details.	create getReservationDetails API	4
		create reservation details UI	2
		test user story	2
WP_4.3	As an admin I want to be able to update listing's amenities documents.	create uploadDocs API	10
		integrate file picker buttons in the UI	3
		test user story	2
WP_5.1	As an admin, I want to check all guest's messages.	create chat api	15
		create getConversationList api	15
		create chat UI	6
		create chat conversation UI	6
		create getCMessagesList api	15
		test user story	5
WP_5.2	As an admin, I want to filter all guest's messages by	create filter api	2
		Integrate filter button in the UI	2
		test user story	2

	reservation, email and listing.		
WP_5.3	As an admin, I want send messages to guests.	create sendMessage api	15
		create SendMessage UI	5
		test user story	2

Table 4 Sprint 1 Backlog

Sprint 1 use case diagram

In this use case diagram, we will show all possible operations that the admin user can perform. We will have a global use case diagram and four other sub diagrams that will explain these use cases: manage profile data, manage reservation, and manage listings and finally the customer care chat service. The following diagram is a global use case diagram in which we present the all the general use cases and after that we will present the four use cases modules that will be detailed as separate diagrams.

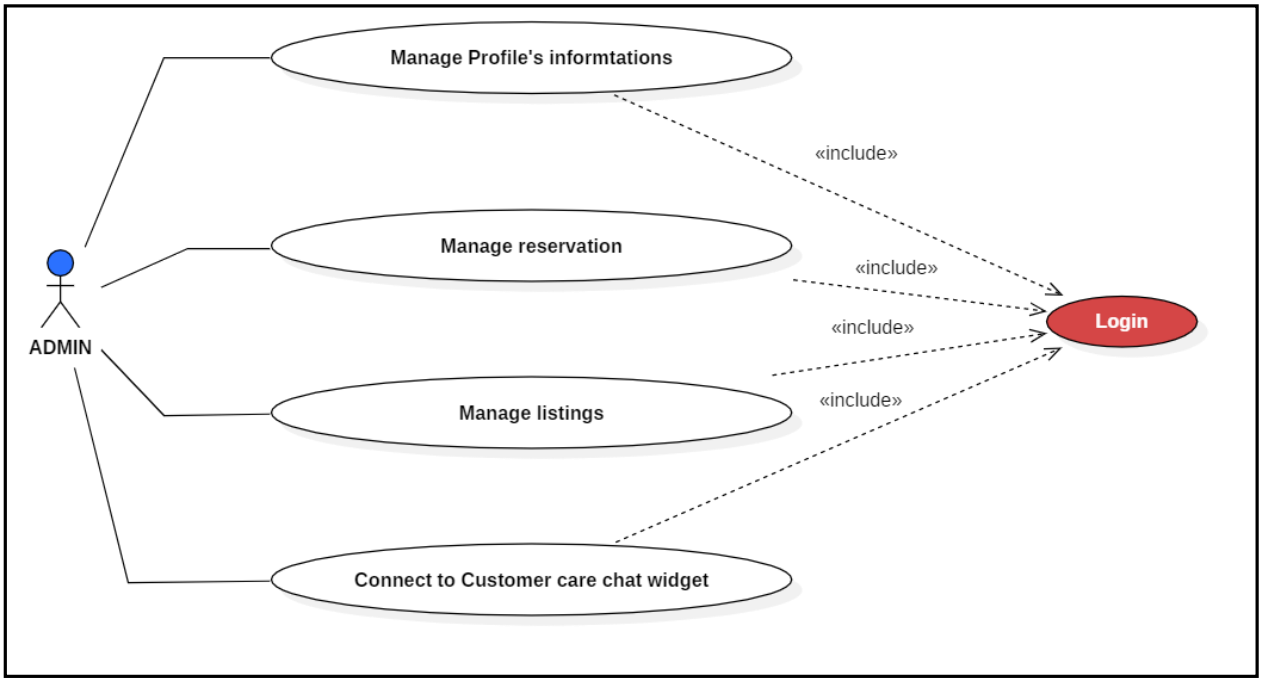


Figure 24 Sprint 1 use case diagram

The next diagram will explain the global use case “manage profile data”.

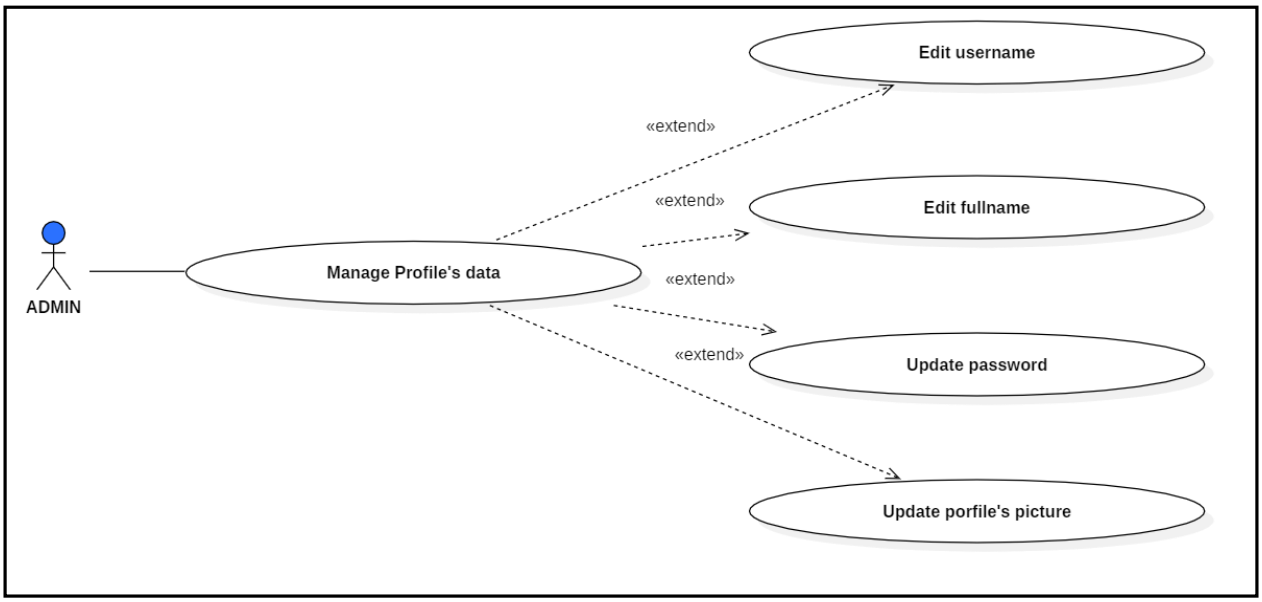


Figure 25 manage profile data use case

The second detailed use case diagram will explain the “manage reservation” use case.

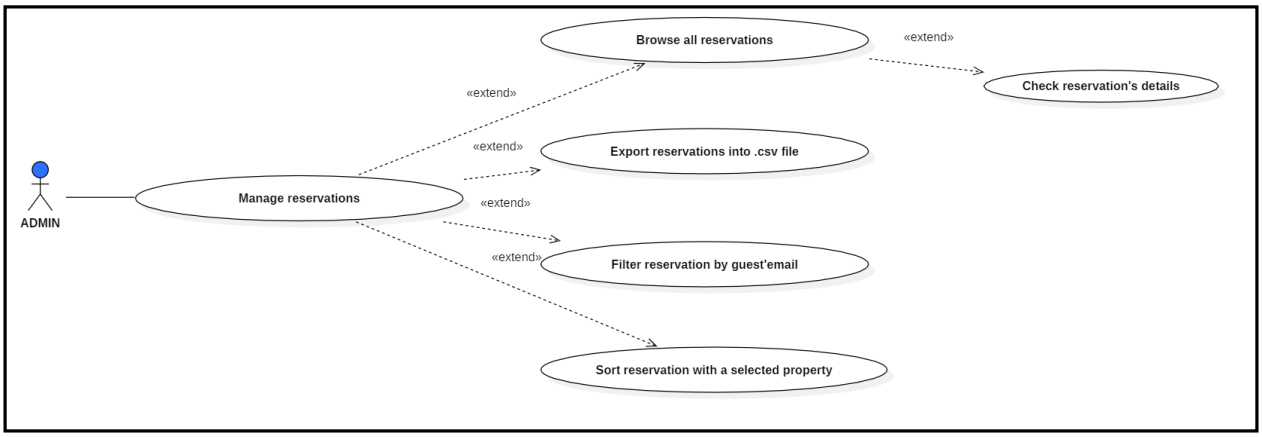


Figure 26 manage reservation use case

The third detailed diagram will explain “manage listings” use case.

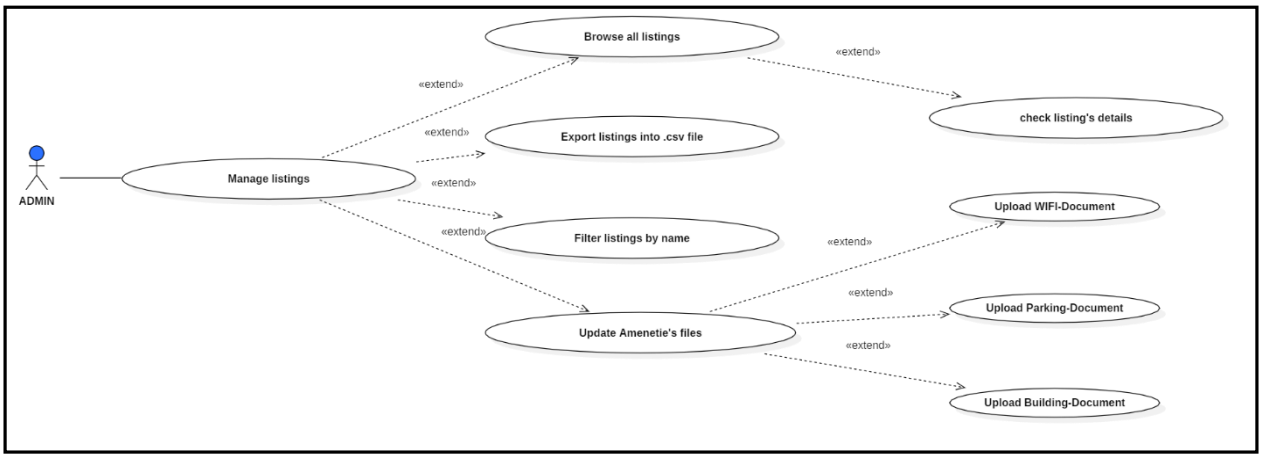


Figure 27 manage listings use case

The fourth and final detailed diagram is explaining how to manage the customer care chat service.

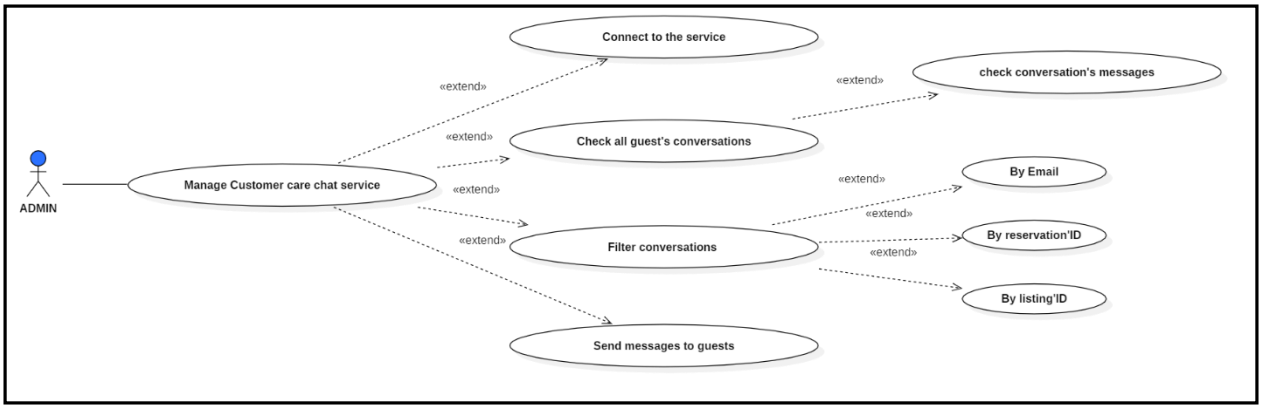


Figure 28 customer care chat use case

Use case 1: Check a reservation’s details

This use case is “**Check a reservation’s details”,** we will provide a textual description and a system sequence diagram.

- **Textual description of “Check a reservation’s details”**

Use case	**Check a reservation’s details**
actors	admin
preconditions	The admin should be logged in.
Post conditions	

Nominal scenario	1. The admin choose reservations from the menu. 2. The system displays the list of reservations. 3. The admin search for a specific reservation using the email of the guest. 4. The system checks if the reservation exists. 5. The admin select the right reservation. 6. The system displays all the details of selected the reservation.
Alternative scenario	• The system can't find that reservation • The system shows an empty list.

Table 5 Textual description of "Check a reservation's details"

- **System sequence diagram**

This system sequence diagram explains all possible interactions between the admin and the web portal when he tries to check a reservation's details.

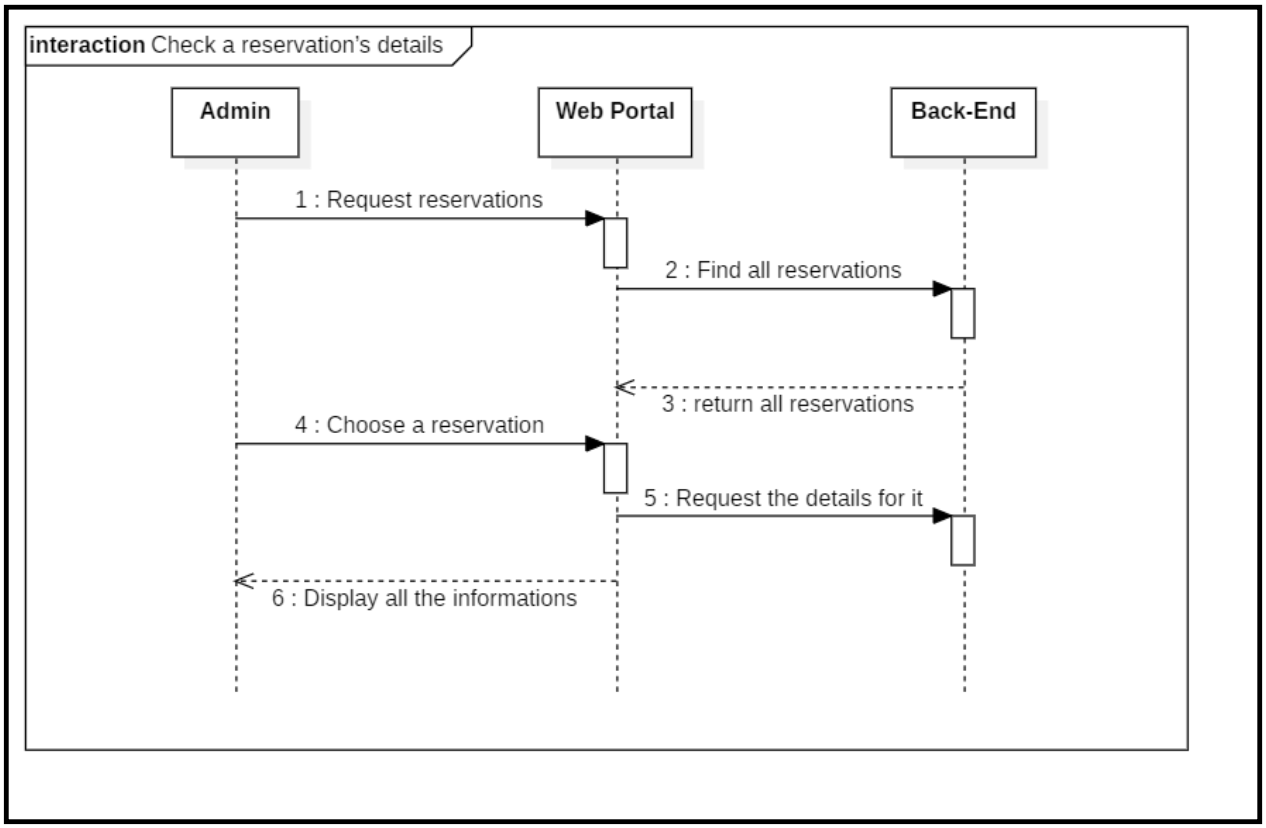


Figure 29 System sequence diagram of check reservation's details

Use case 2: Send a response for a specific reservation

The second studies use case is a feature developed in the administration web portal in which an admin can reply a guest using the customer care chat widget for a specific reservation.

- **Textual description of "Send a response for a specific reservation"**

Use case	**Send a response for a specific reservation**
actors	admin
preconditions	The admin should be logged in.
Post conditions	
Nominal scenario	1. The admin choose chat from the menu. 2. The system displays the list of conversations. 3. The admin search for a specific conversation using the reservation's

	id, the listing's id or the email of the guest. 4. The system checks if the conversation exists. 5. The admin select the right conversation. 6. The system displays all messages related to that conversation. 7. The admin answer the guest. 8. The system notify the guest.
Alternative scenario	• The system can't find that conversation. • The system shows an empty list.

Table 6 Textual description of "Send a response for a specific reservation"

- **System sequence diagram**

The following system sequence diagram shows all possible interactions between the admin and the web portal in order to answer a guest's message from the customer care chat.

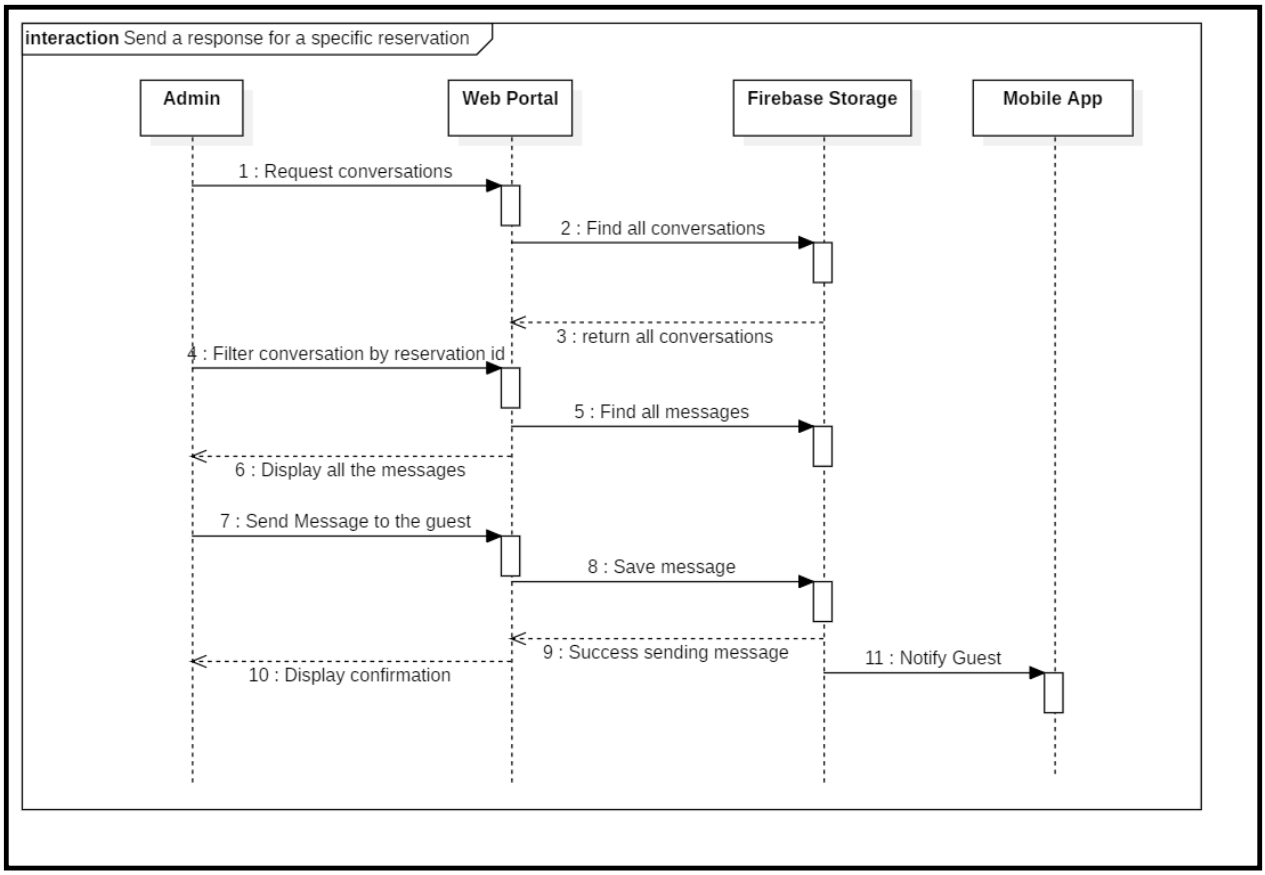


Figure 30 System sequence diagram of Send a response for a specific reservation

Analysis class diagram

The analysis class diagram will illustrate the classes of the system and their inter-relationships. For each class we will provide the needed attributes and also the operations.

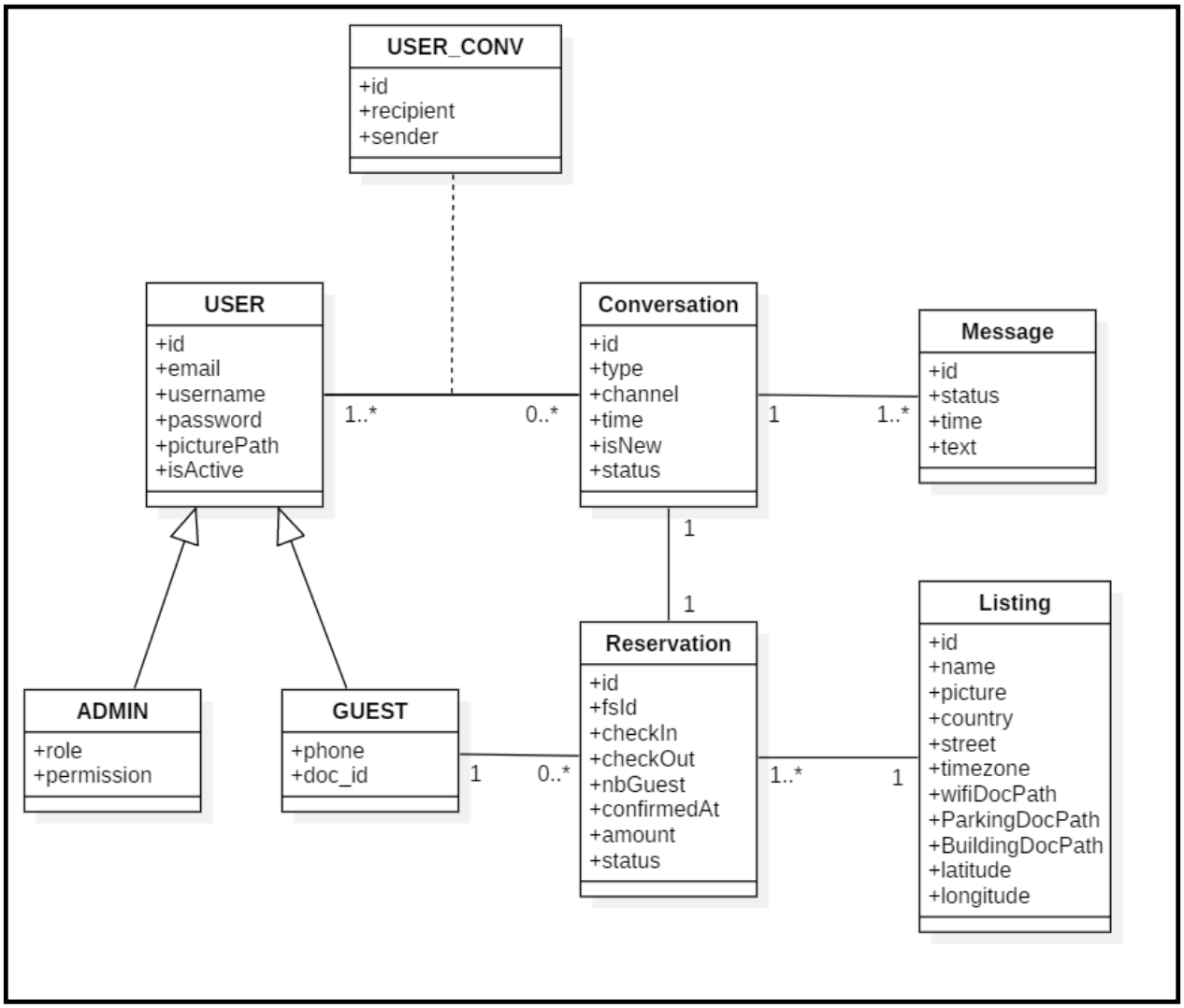


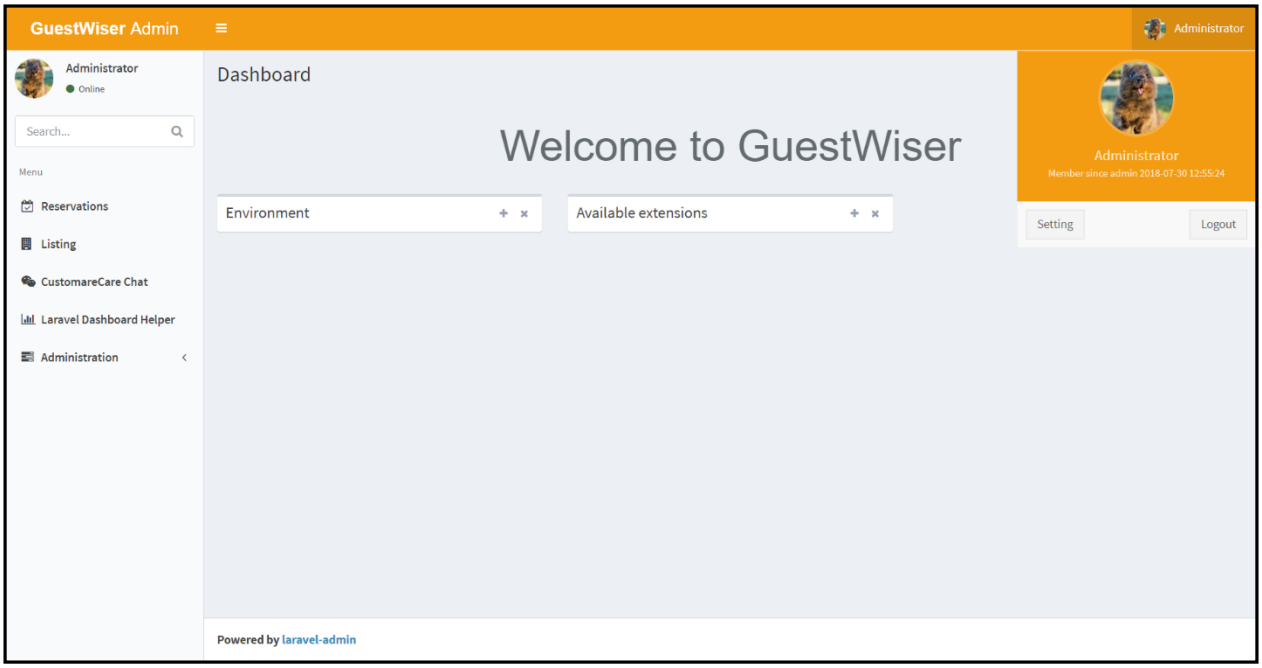
Figure 31 Sprint 1 analysis class diagram

Sprint review

After performing the needed tests for each task in the web portal and finishing the development process, we had another sprint meeting. During this meeting a full product demonstration is made for the product owner so we can get his feedback. We present next the graphical interfaces of our solution.

- **Home Interface**

The following interface is the home page of the administration portal, as we can see it includes the basic information of the admin user, the navigation menu and the top left menu in order to update the user's credentials etc.



- **Listing Interface**

The next interface shows the list of all integrated apartments in GuestWiser from which the admin could filter the list using the name or exporting it in .csv file as shown in the second screenshot.

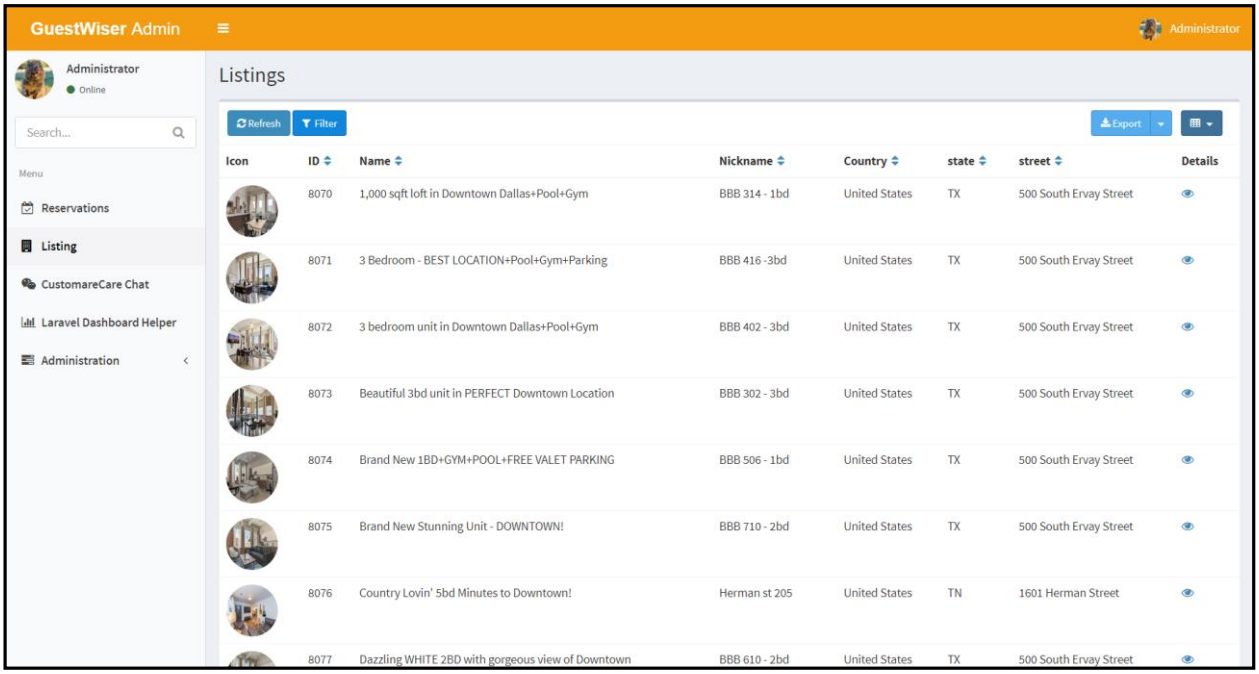


Figure 32 Listing Interface

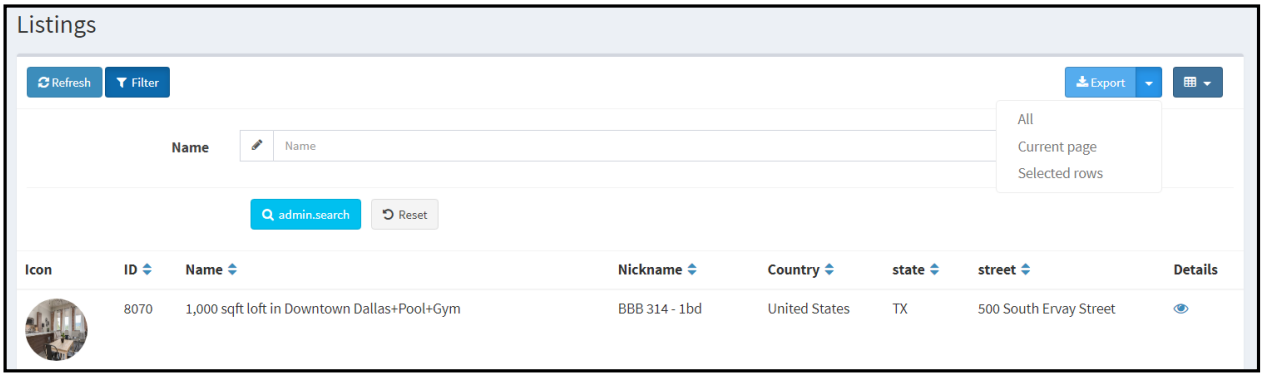


Figure 33 Listing item Interface

The next interface is all about the listing's details where an admin can update the files of the parking, the Wi-Fi and the building rules for the apartments also he can check all the basic information about it. It shows the address on a map to facilitate locating the apartment.

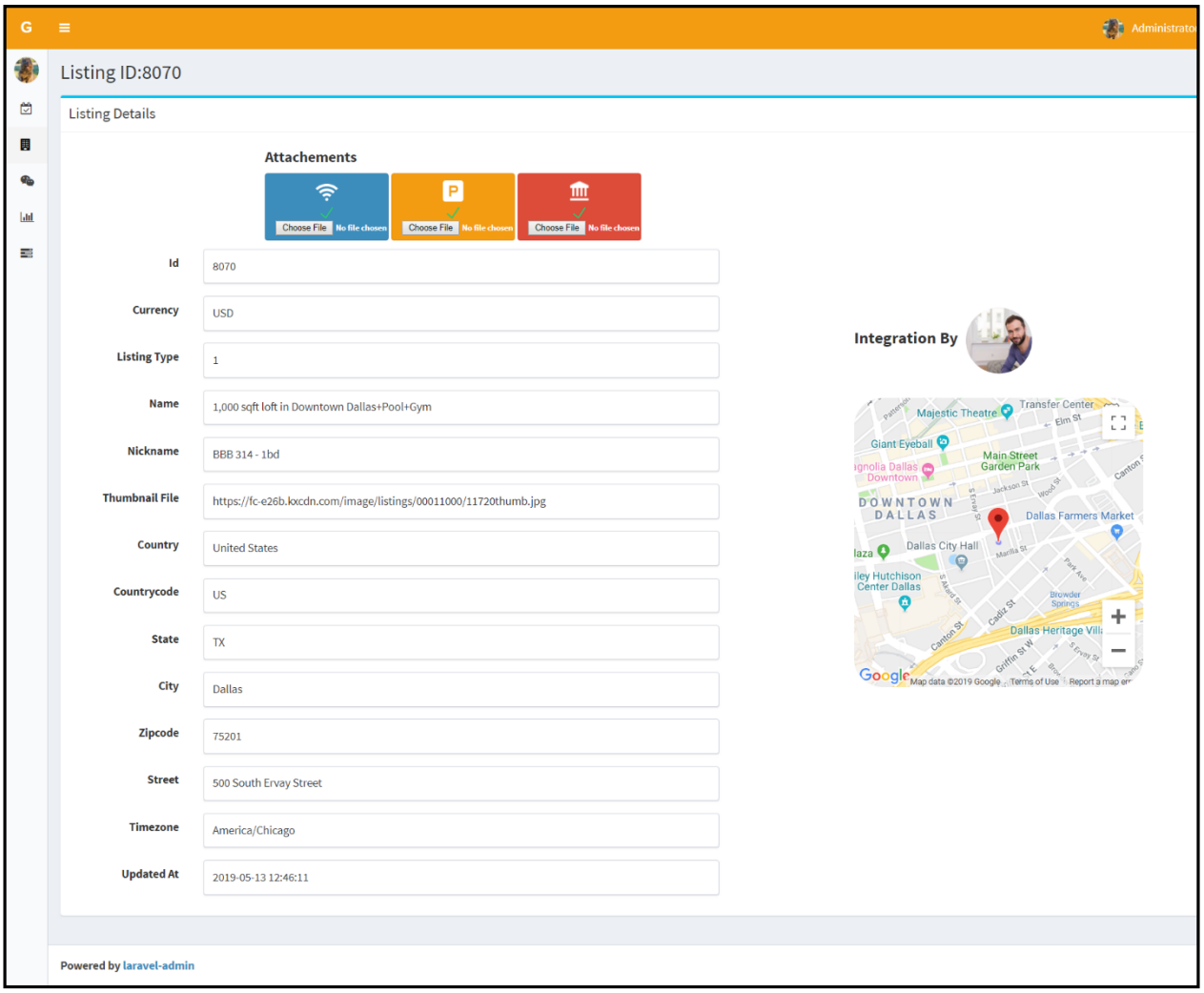


Figure 34 Listing details interface

- **Reservation Interface**

This interface helps an admin to control all the reservations made in fantastic Stay from which the admin could filter the list using the name or exporting them in .csv file as shown in the second screenshot.

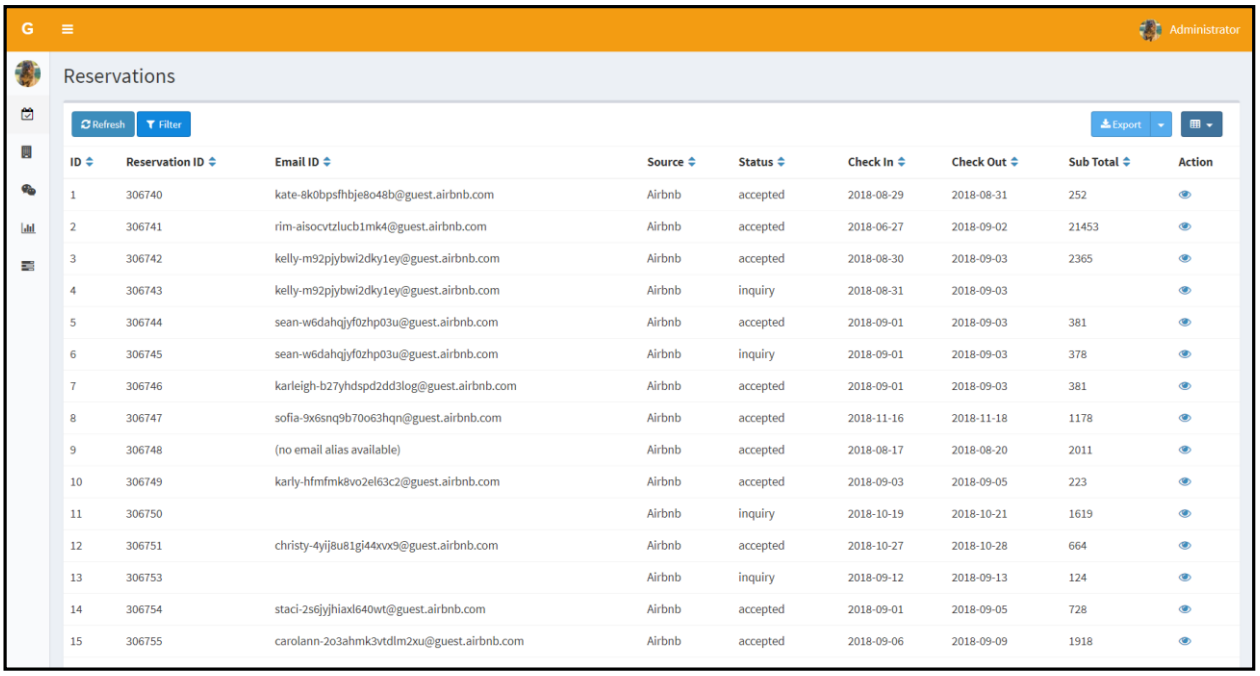


Figure 35 Reservation Interface

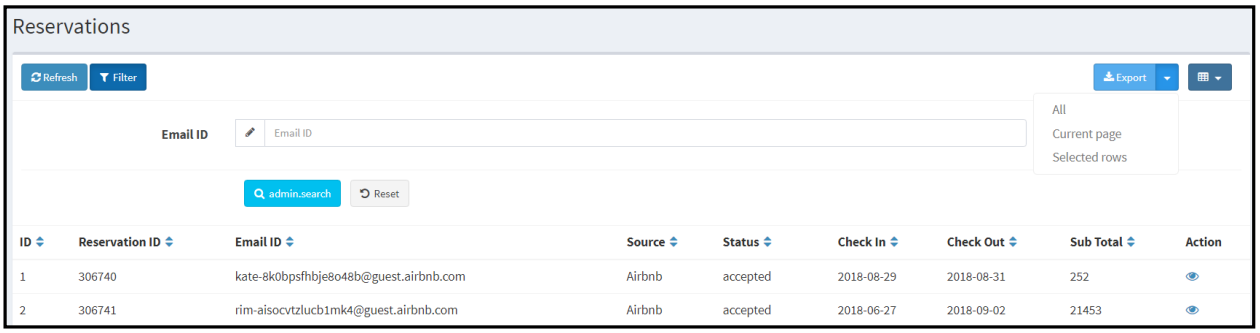


Figure 36 Reservation item Interface

The next interface is about the reservation's details where an admin can check all the information about a specific reservation.

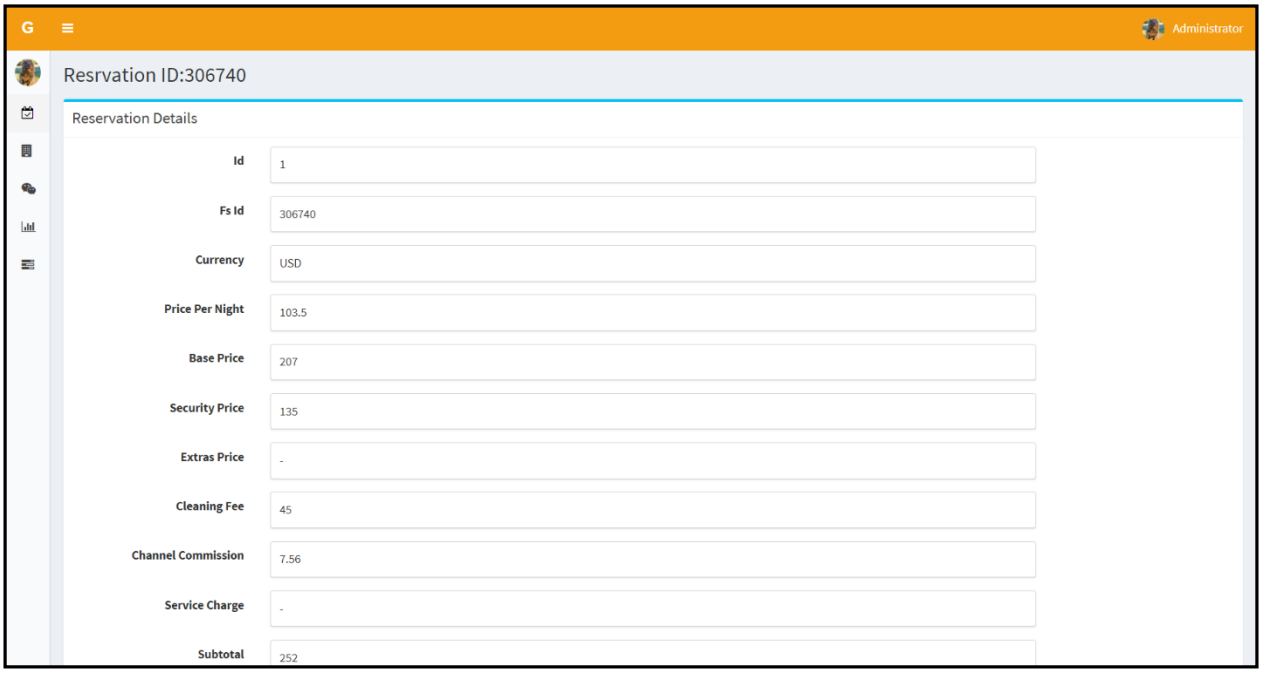


Figure 37 Reservation details Interface

- **Customer care chat Interface**

This interface represents the chat widget for the customer care service, it includes all the conversations received from guest. Every conversation has a sender name, message's date, and its related reservation. The admin could filter the conversations using listing's id, guest's email and also the reservation's id as demonstrated in the second screenshot.

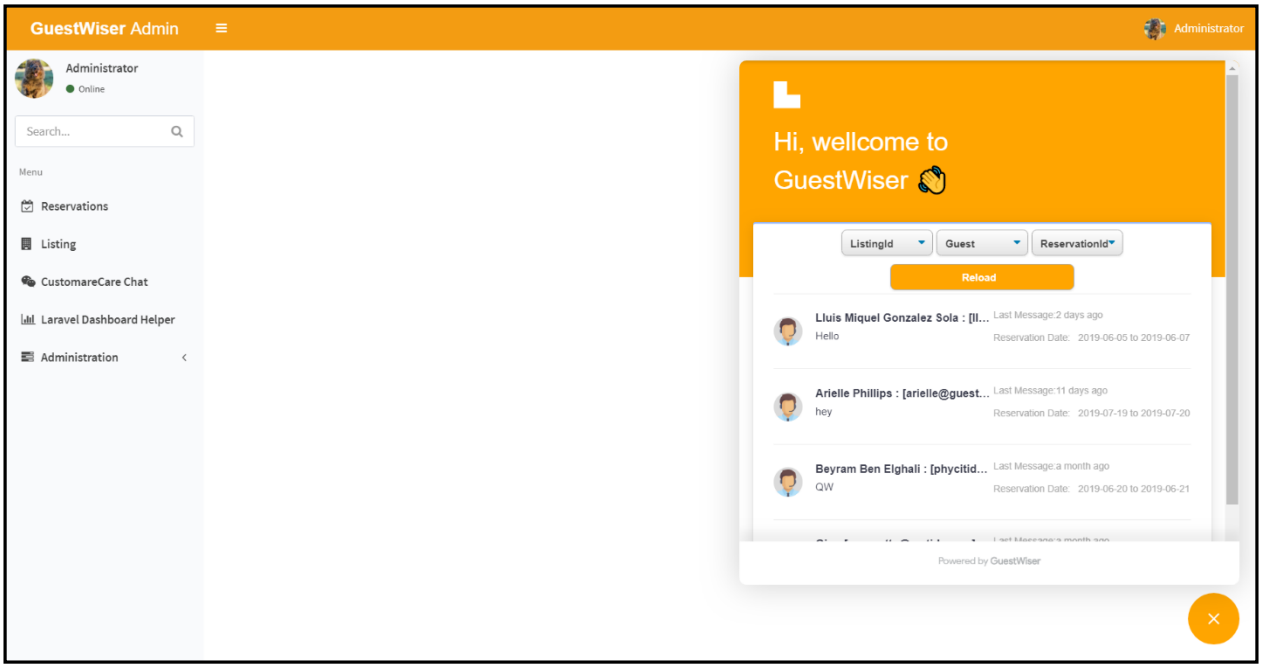


Figure 38 Customer care chat Interface

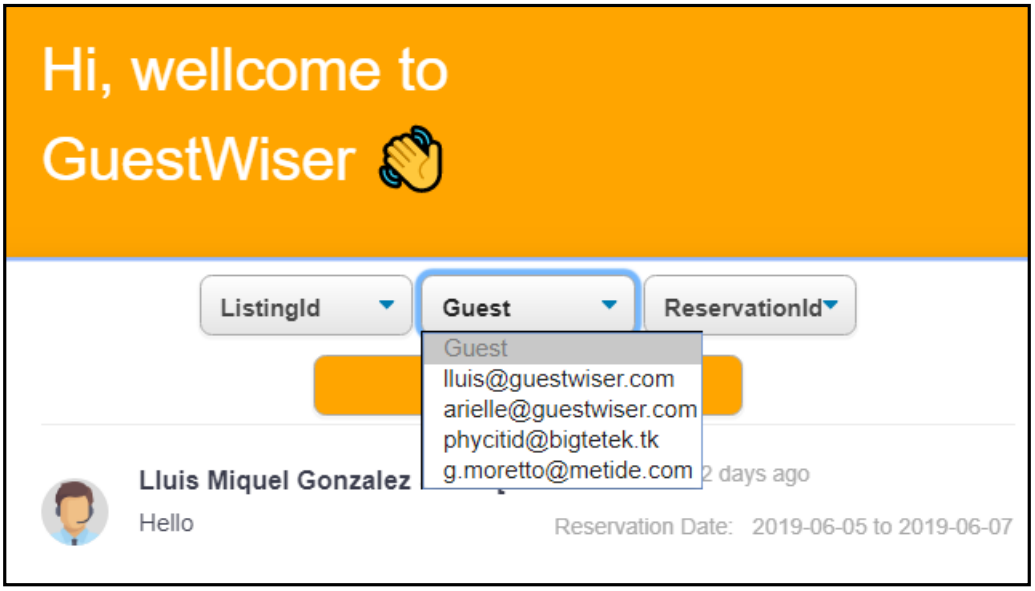


Figure 39 Customer care chat conversation filters

The following screenshot shows the list of the messages between a guest and an admin.

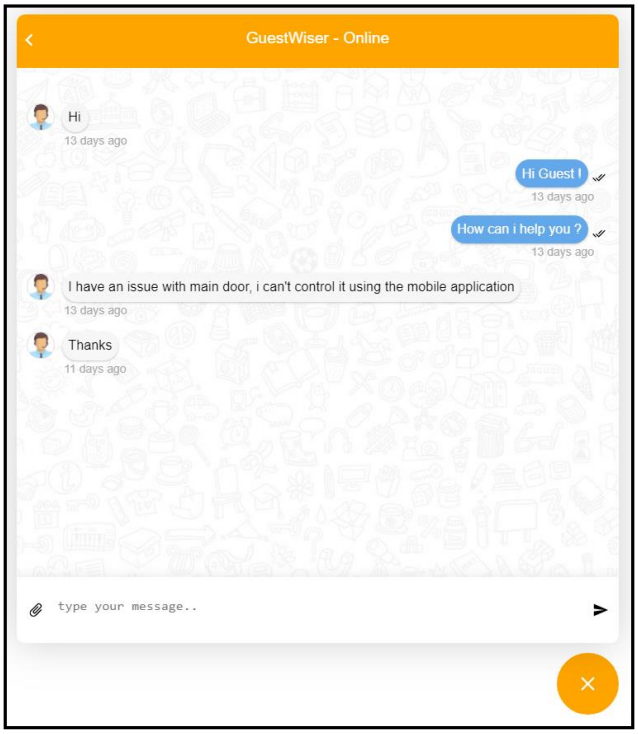


Figure 40 Conversation interface

- **Setting Interface**

The next and final screenshot shows the ability to change the basic information of the admin user like updating the picture, editing password etc.

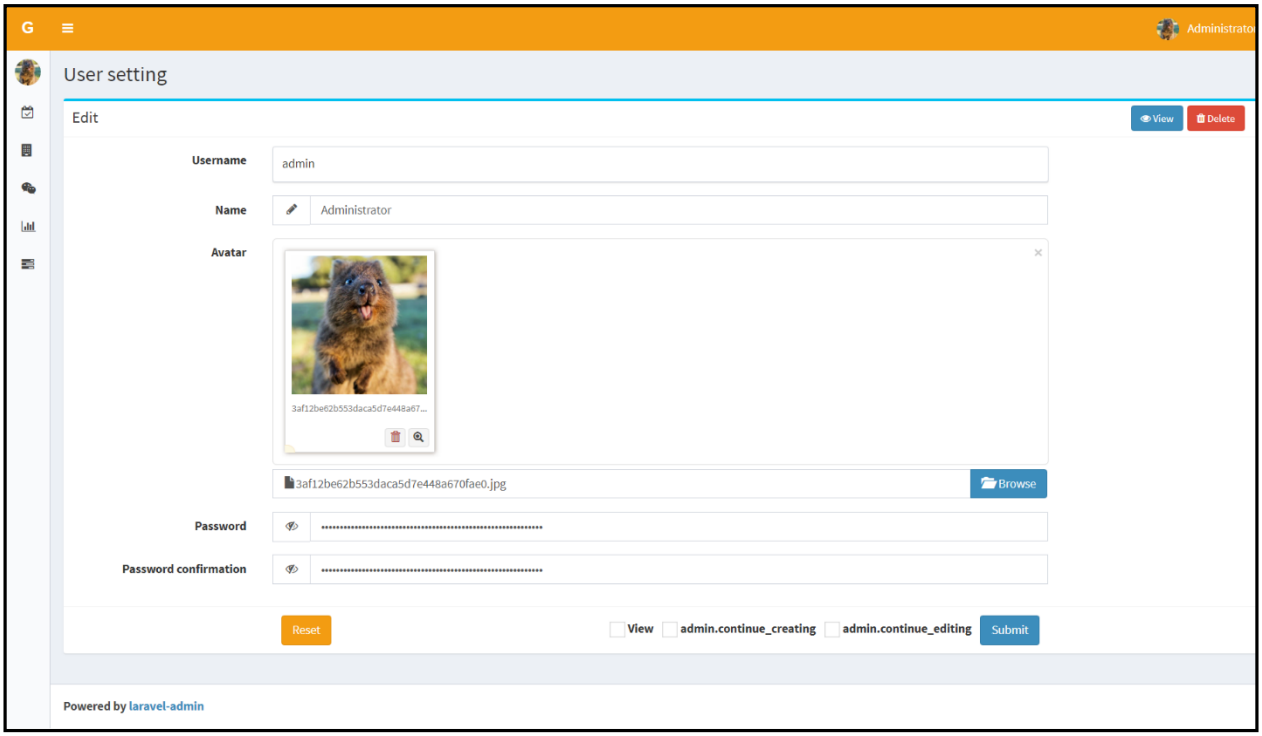


Figure 41 Setting Interface

Burn down chart

After finishing our tasks, the final phase of the sprint is to extract a sprint burndown chart that shows the whole time we spent on the development of the needed tasks, so we can analyze our work and check what went wrong during the sprint.

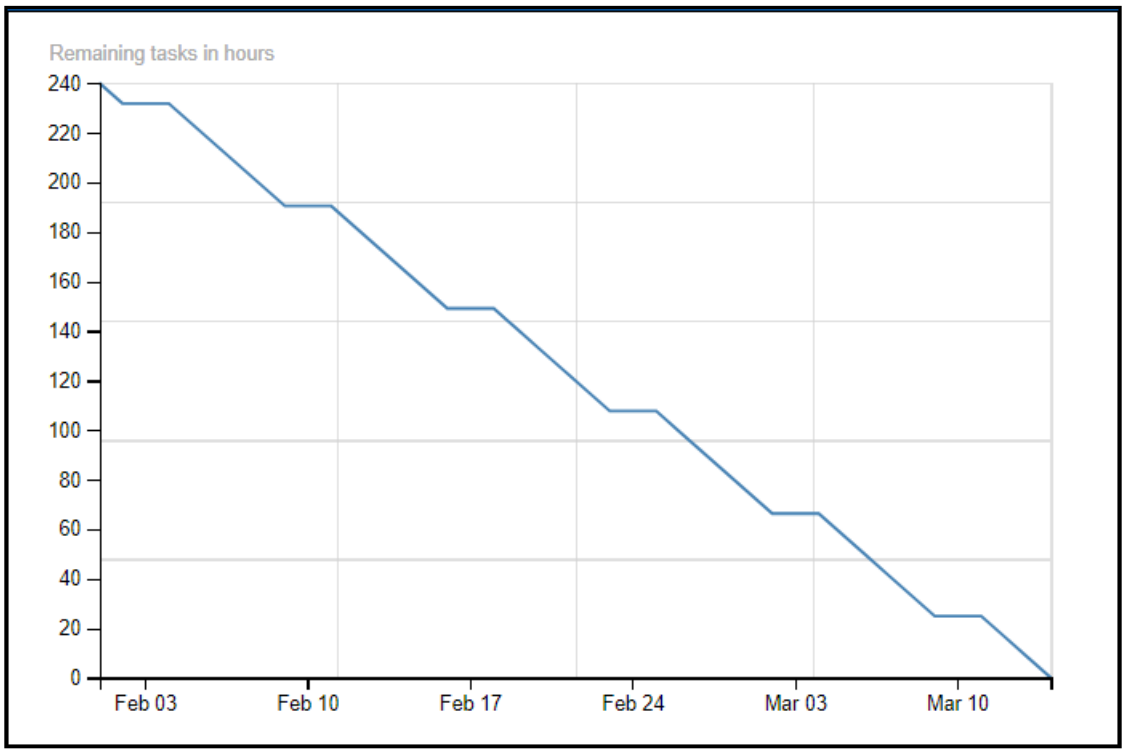


Figure 42 Sprint 1 Burn down chart

As a result of analyzing this chart, we can see that the web portal solution was delivered two days earlier than expected. That was a very good kickoff since we created a stable and efficient web application. However, during the first sprint backlog we expected that the tests of the tasks will take longer than they actually did, by that we gained too much time.

Conclusion

In this chapter which presents the first sprint of the project, we have designed, built the administration web portal. We have prepared all what customer need in in an efficient and stable. In the next part, we continue working on the next sprints for the development of the mobile application in both iOS and Android version.

CHAPTER 5: SPRINT 2 “iOS”

Introduction

In this chapter, we will present the second sprint of this project which is mainly related to the development of the iOS application for GuestWiser guests and developing the tasks with priorities between 50 and 100 (100 is the highest). First, we will describe all the user story and its needed tasks side by side with the class diagram and a detailed sequence diagrams for certain delicate use cases, then we will start developing our iOS. In the end of this sprint, we will elaborate a total review by elaborating a burn down chart and presenting the UI interfaces of the application.

Functional specifications

During The functional specifications we will elaborate a sprint backlog and the use case diagram for a specific use cases.

Sprint Backlog

The following table is the second sprint backlog dedicated to the development of the iOS application for GuestWiser guests, it will contains all user stories, their related tasks and also a time estimation for each one, knowing that this a 2 months sprint contains almost 350 working hours.

ID	User story	Task	Time(h)
MA_IOS_1.1	As a guest I want to Create an account.	create user register API	4
		create register UI	7
		develop register logic	4
		test user story	2
MA_IOS_1.2	As a guest I want to login from the application.	create user login API	3
		create login UI	4
		develop login logic	4
		test user story	2
MA_IOS_2.1	As a guest I want to consult all my reservations.	create getReservations API.	4
		create home UI	6

		develop logic	4
		test user story	2
MA_IOS_2.2	As a guest I want to filter reservations by status.	develop logic	3
		integrate filter button in the UI	2
		test user story	2
MA_IOS_2.3	As a guest I want to check reservation details.	create getReservationDetails API	5
		create details UI	5
		develop logic	5
		test user story	2
MA_IOS_2.4	As a guest I want to download amenity's documents.	create getDocs API	8
		Integrate buttons in the UI	2
		develop logic	5
		test user story	2
MA_IOS_2.5	As a guest I want to open the main building door.	create openMainDoor API	5
		Integrate button in the UI	1
		develop logic	4
		test user story	2
MA_IOS_2.6	As a guest I want to open/close the apartment's door.	create openListingDoor API	8
		create Reservation's details UI	5
		develop logic	4
		test user story	2
MA_IOS_2.7	As a guest I want to check/change the thermostat's temperature.	create getTmp API	4
		create changeTmp API	4
		Create UI components	3
		develop logic	3
		test user story	2
MA_IOS_2.8	As a guest I want to use Maps in order	Integrate button in the UI	1
		develop logic	6

	to get direction to the apartment.	test user story	2
MA_IOS_3.1	As a guest I want to send a message to the customer care service.	create sendMessage API	10
		Create chat UI	5
		develop logic	10
		test user story	4
MA_IOS_3.2	As a guest I want to check all my messages with the customer care service.	create getMessage API	10
		Create message UI	5
		create chat UI	5
		develop logic	7
		test user story	2
MA_IOS_3.3	As a guest, I want to get notification if I receive a message.	Implement push notification	8
		test user story	3
MA_IOS_4.1	As a guest, I want to invite other guests to my apartment.	create inviteGuest api	6
		create Invitation UI	5
		Develop logic	5
		test user story	3
MA_IOS_4.2	As a guest, I want to remove guest‘s invitations	create removeGuest api	4
		Integrate button in the UI	2
		develop logic	5
		test user story	4
MA_IOS_5.1	As a guest, I want to browse all GuestWiser online deals using maps	create getDeals api	4
		Create Deals UI	5
		Maps integration	5
		develop logic	3
		test user story	2
MA_IOS_5.2		create getDetail api	3
		Create detail UI	3

	As a guest, I want to check listing's details.	develop logic	3
		test user story	1
MA_IOS_5.3	As a guest, I want to book an apartment using an online payment service.	Create setBooking API	5
		payment system integration	16
		develop logic	5
		Create payment UI	5
		test user story	5

Table 7 Sprint 2 Backlog

Sprint 2 use case diagram

In this use case diagram, we will show all possible operations that the guest user can perform. We will have a global use case diagram alongside with three other sub diagrams that will explain these use cases, manage reservations, guest's invitation and finally the online deals. The following diagram is a global use case diagram in which we listed all the possible use cases, after that the four use cases modules that will be detailed as separate diagrams.

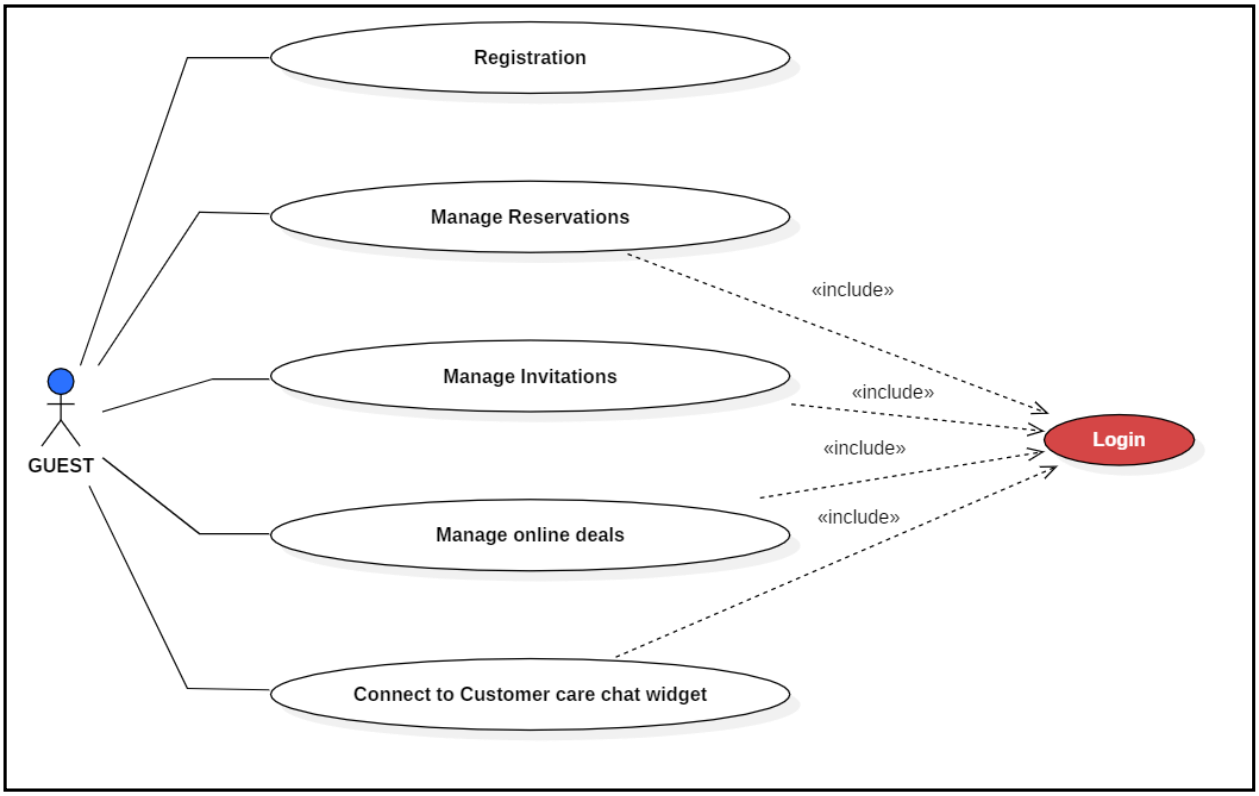


Figure 43 Sprint 2 use case diagram

The next diagram will explain the global use case “manage reservations”.

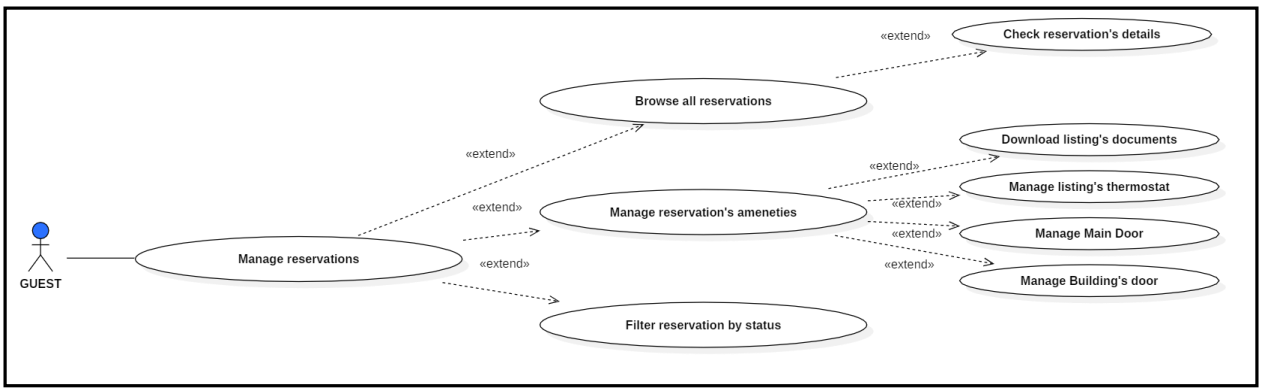


Figure 44 manage reservations use case

The second detailed use case diagram will explain the “manage invitations” use case.

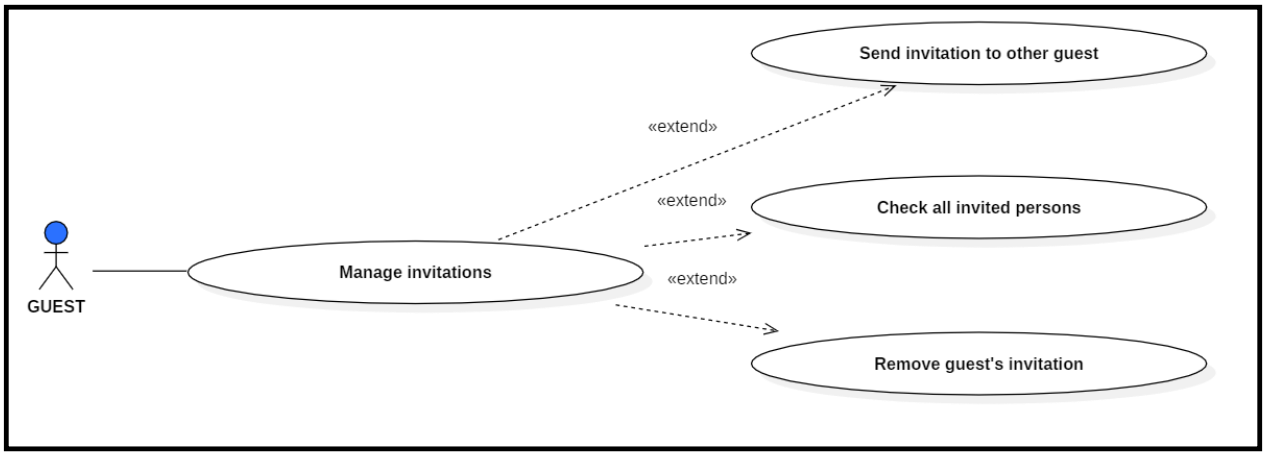


Figure 45 Manage invitations use case

The third and final detailed diagram is explaining how to manage online GuestWiser deals.

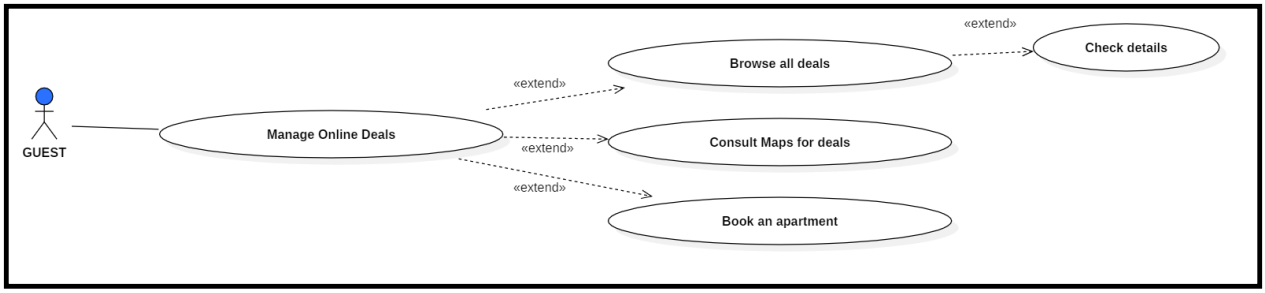


Figure 46 Manage online deals use case

Use case 1: Create an account

This use case is "**Create an account"** which is a feature available only on the mobile application where the client can subscribe to GuestWiser by creating an account.

Textual description of "create an account"

Use case	**Create an account**
actors	Guest
preconditions	
Post conditions	Auto login
Nominal scenario	1. The client fill the registration form. 2. The client attach a picture of his ID. 3. The system check that the account exist.. 4. The system creates a new use account.

	5. The mobile application shows a confirmation notification. 6. The system send an email to confirm the account. 7. The user confirm his account.
Alternative scenario	• The system check that the account exist. • The system shows an error that the username or email exists.

Table 8 Textual description of "create an account"

System sequence diagram

This system sequence diagram shows all the possible interactions between a user and the mobile application when he start to create an account.

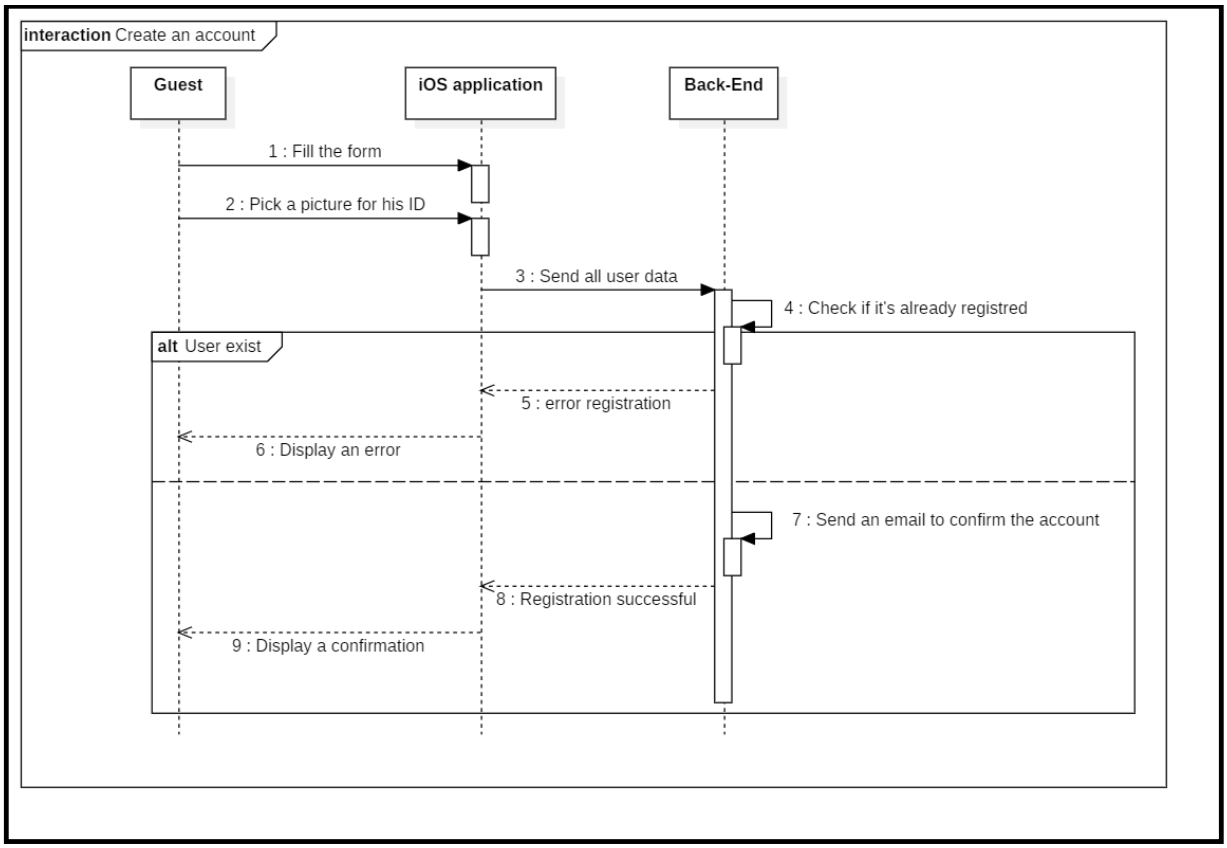


Figure 47 System sequence diagram of create an account

Use case 2: Check all active reservations

This use case is "**Check all active reservations",** we will provide a textual description and a system sequence diagram.

- **Textual description of "Check all active reservations"**

Use case	**Check all active reservations**
actors	Guest
preconditions	The admin should be logged in.
Post conditions	
Nominal scenario	1. The guest choose reservations from the menu. 2. The system displays the list of reservations. 3. The guest click on the filter button to get active reservation. 4. The application displays only active ones.
Alternative scenario	• The application can't find any active reservation. • The application shows an empty list.

Table 9 Textual description of "Check all active reservations"

- **System sequence diagram**

This system sequence diagram explains all possible interactions between the guest and the web portal when he tries to check the active reservation reservations.

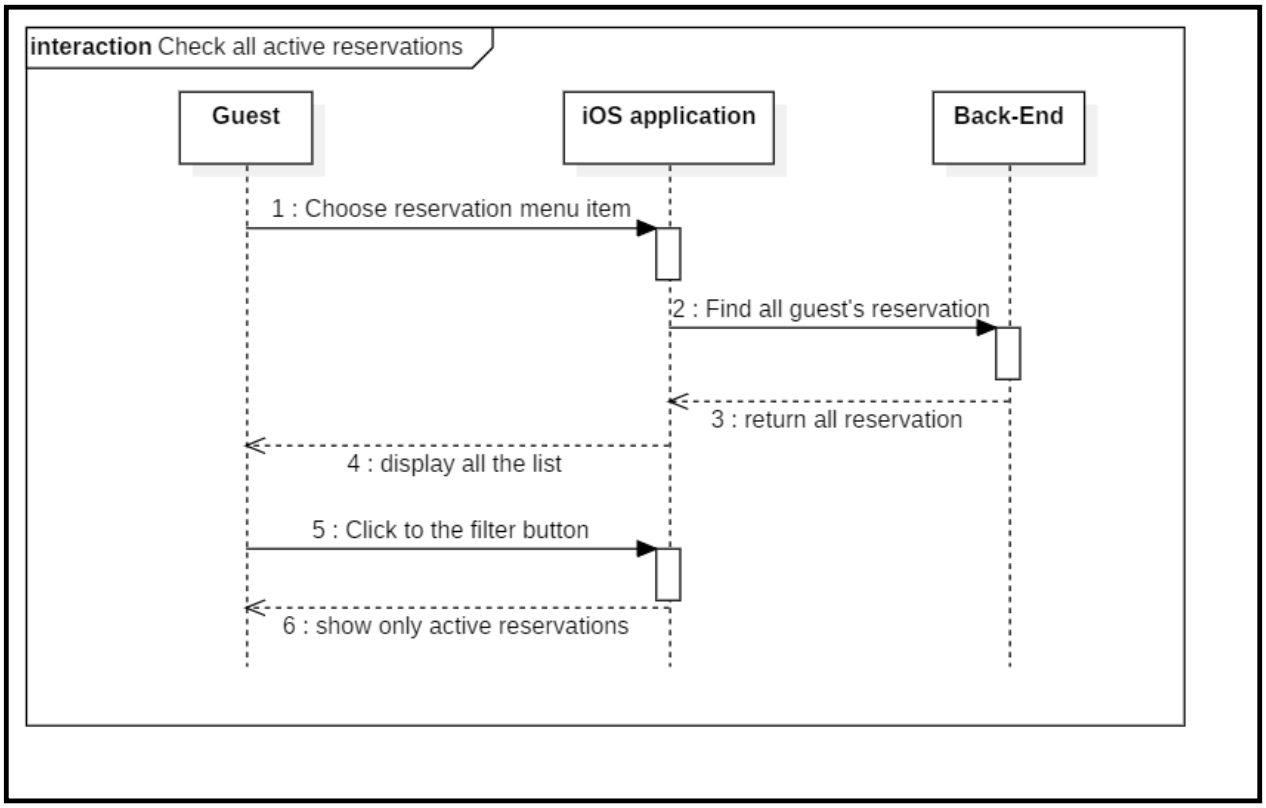


Figure 48 System sequence diagram of check active reservation

Analysis class diagram

The analysis class diagram will illustrate the classes of the iOS application. It will include all the new classes and the modification for the old ones in order to achieve our goal during this second sprint.

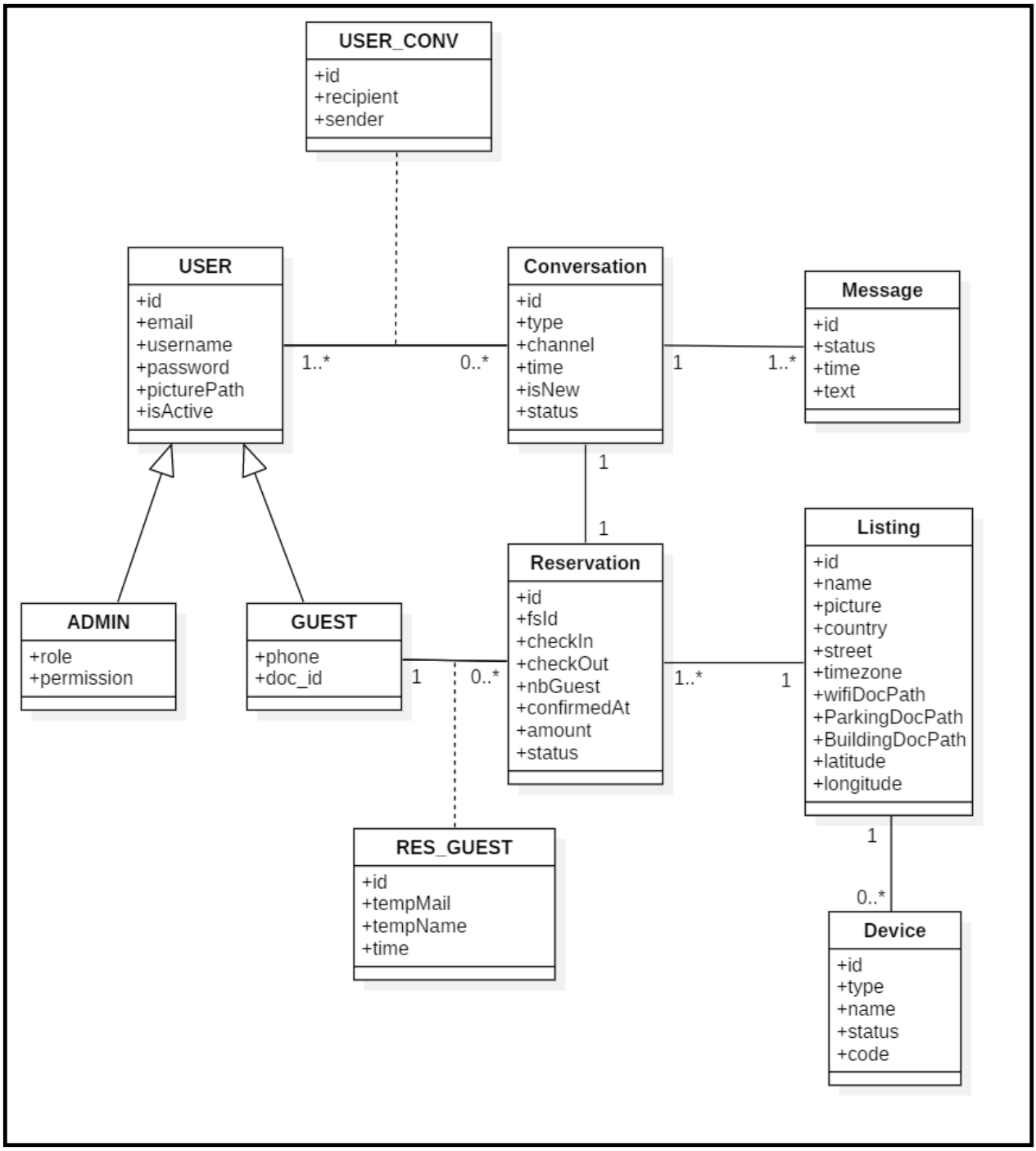


Figure 49 Sprint 2 Analysis class diagram

Sprint review

After performing the needed tests for each task in the iOS application and finishing the development process, we made a demonstration and approved this sprint. The next screenshots demonstrate the graphical interfaces of our solution. We will show only the screenshots for the studied uses cases in this chapter in order to avoid repetition in the review. The following

screenshots are the splash screen, the login interface where the user write his credentials in order to access to his account and finally the menu display which allow the navigation between the features in the application by selecting one item.

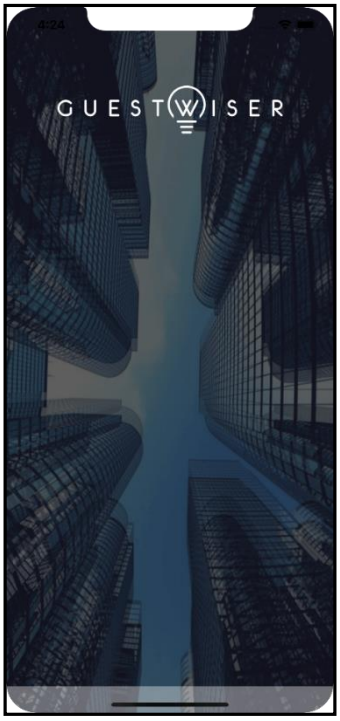


Figure 50 Splash screen

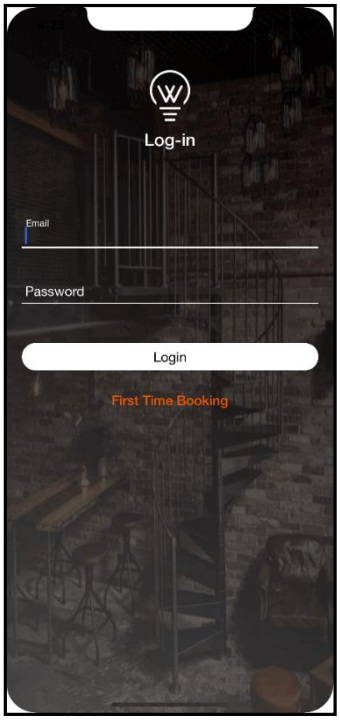


Figure 51 Login screen

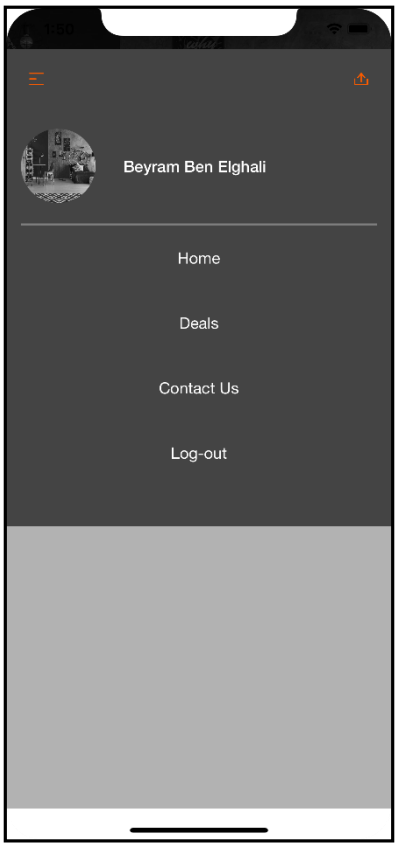


Figure 52 menu UI

- **Reservation :**

The next screenshot demonstrates the reservation feature where the user could browse all his reservations and filter them in order to check all the active bookings. Every Item has a listing's name, dates etc.

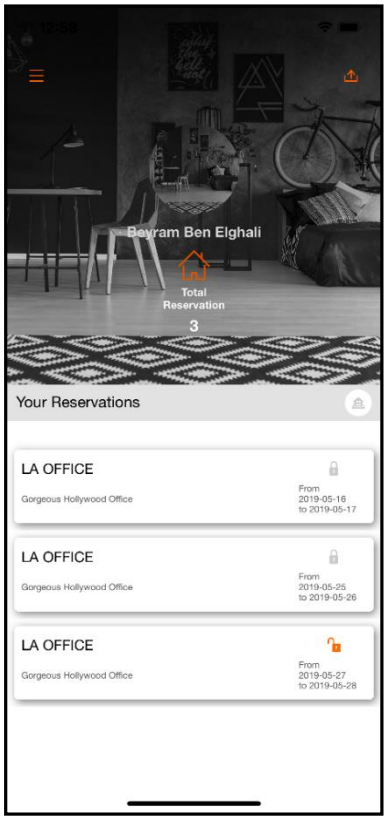


Figure 53 Reservation list

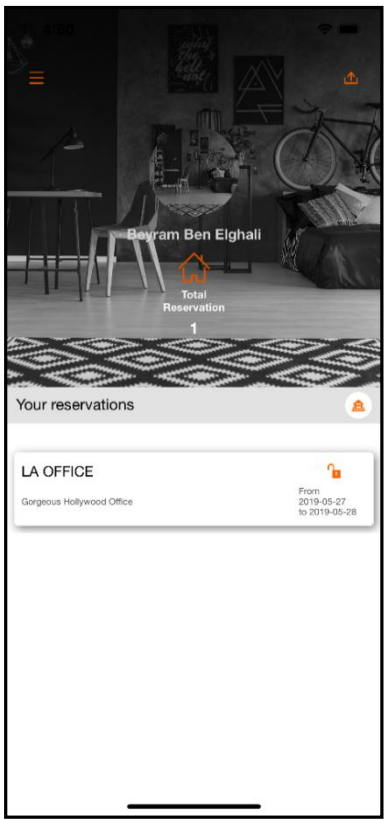


Figure 54 Active reservation list

- **Invitation :**

These interfaces shows how a guest could invite other guests by completing a form with the guest's data and allow them to create an account in GuestWiser in order to give them access to his reservation and manage everything about it.

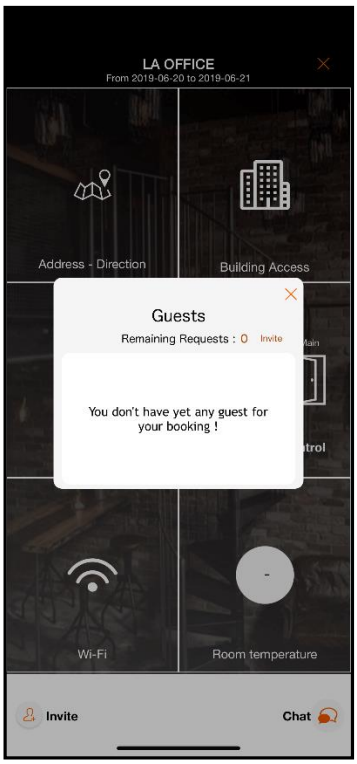


Figure 55 Invitation list

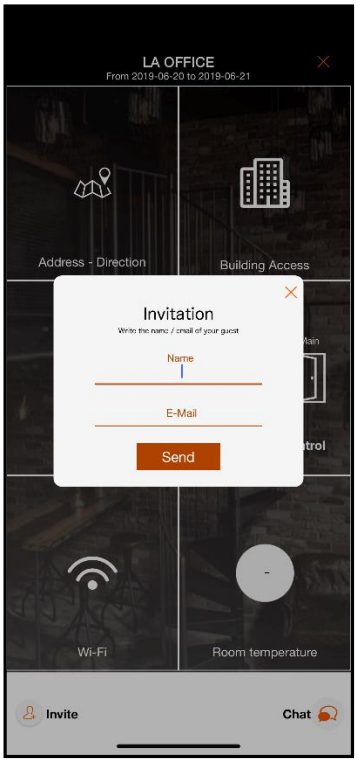


Figure 56 Send invitation form

The owner could see all his guests and could not send invitation more than the indicated number during the booking on fantastic stay.

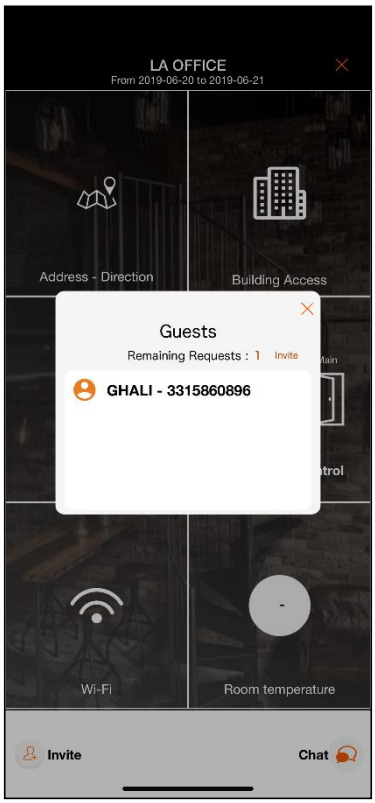


Figure 57 Invitation list interface

After sending an invitation the guest will receive an email and ask him to join GuestWiser.

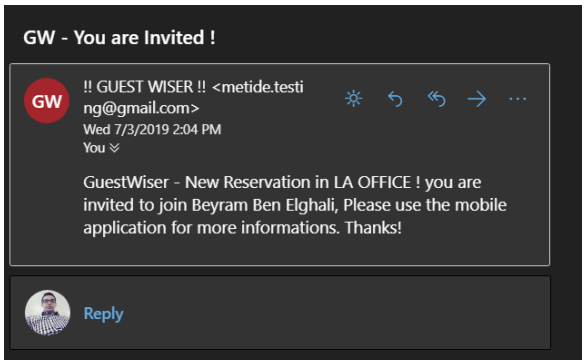


Figure 58 confirmation e-mail of the invitation

The owner of this reservation have the right to remove the guests as well.

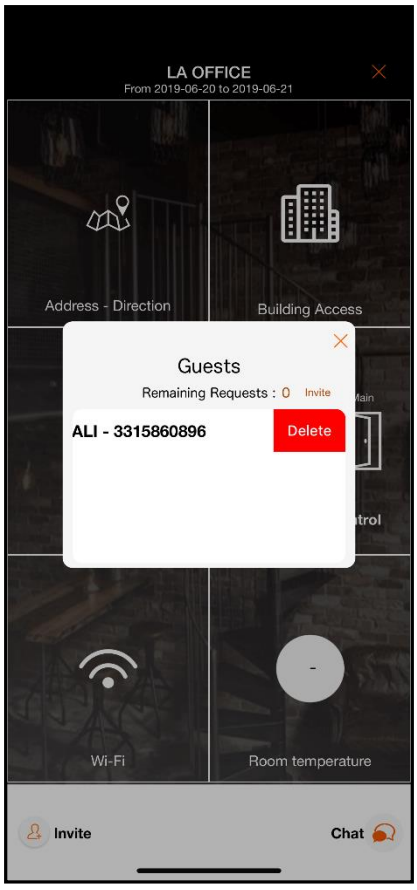


Figure 59 Remove invitation online interface

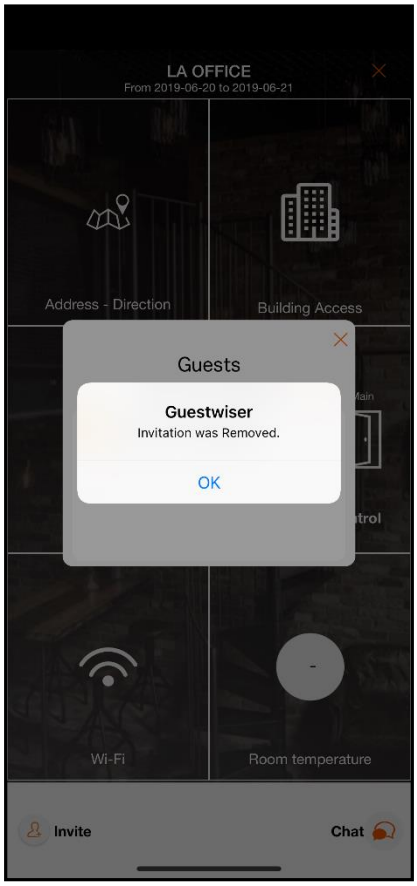


Figure 60 dialog after removing invitation

- **Deals :**

The last section of this review display all the online deals available in GuestWiser. All the items are presented as a marker on the map and by clicking on them, the application will show the list of the apartment available in that area.

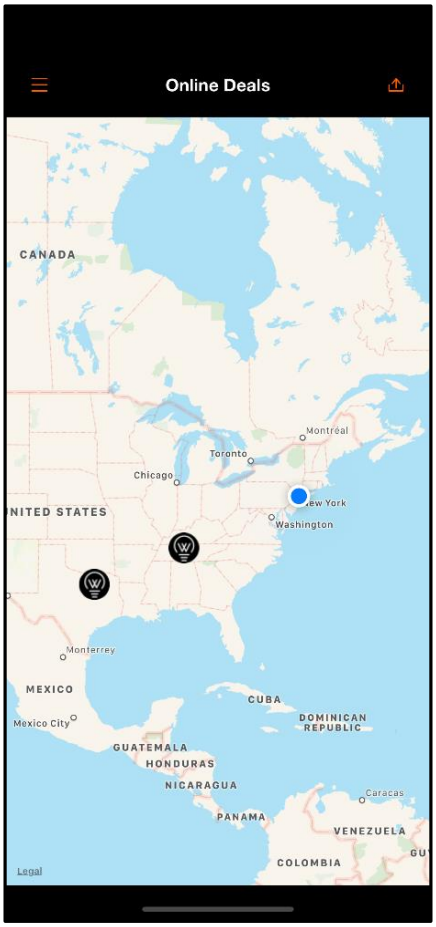


Figure 61 online deals map interface

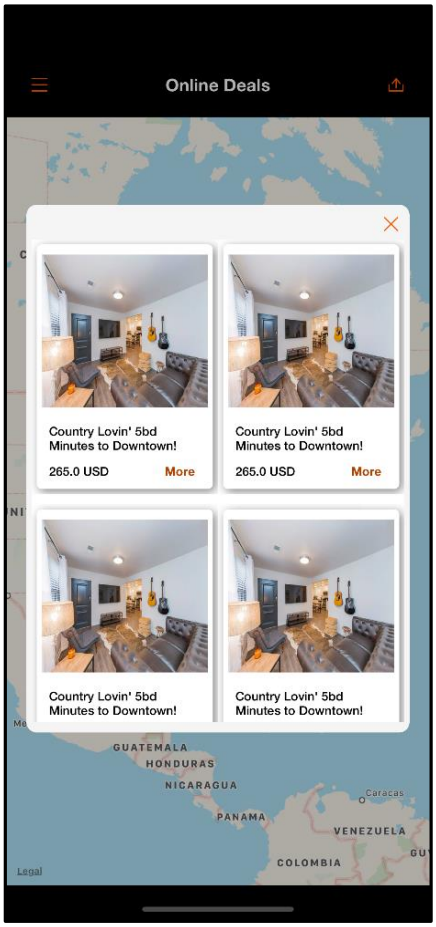


Figure 62 deals interface

Burn down chart

After finishing our tasks, we extracted the Sprint 2 burndown chart that illustrates the time spent in the development of the tasks regarding the expected timeline

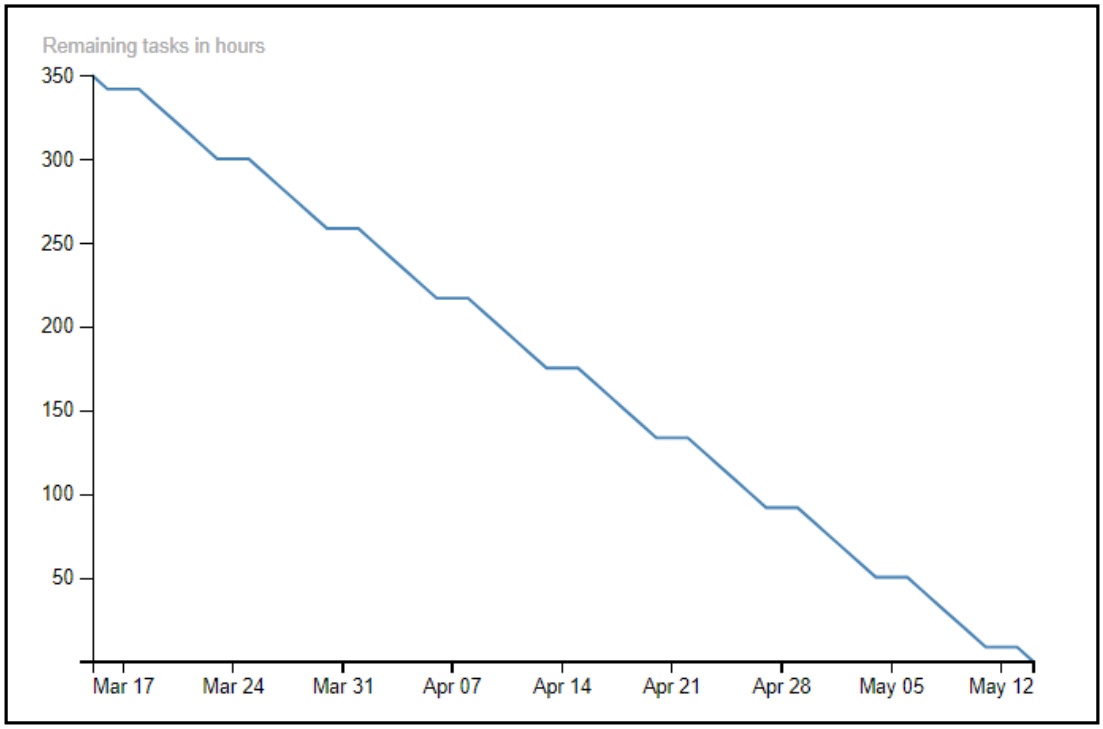


Figure 63 Sprint 2 Burn down chart

As a result of analyzing this chart, we can see that iOS application was delivered four days earlier than expected. This is a great step forward since the iOS application was published in the app store and it becomes public for GuestWiser guests to use it.

Conclusion

In this chapter which presents the second sprint of the project, we have built, published the iOS application for the guest's users. We have prepared all what customer need in in an efficient and stable. In the next and the last part, we continue working on the next sprint for the development of the android version of our application.

CHAPTER 6: SPRINT 3 “Android”

Introduction

In this chapter, we will present the third sprint of this project which is mainly related to the development of the Android application for GuestWiser guests and developing the tasks with priorities between 50 and 100 knowing that 100 is the highest. The same as the first and second sprint, first we are going to elaborate the functional specifications including the sprint use case diagram, the sprint backlog side by side with the class diagram and a detailed sequence diagrams for certain use cases, then we will start developing our Android application. Finally, we will have a sprint review after the realization and retrospective to evaluate the work.

Functional specifications

During The functional specifications we will elaborate a sprint backlog and the use case diagram for a specific use cases for our Android application.

Sprint Backlog

The next table is the third sprint backlog dedicated to the development of the Android application for GuestWiser guests, it will be quite similar to the backlog related to the development of the iOS application.

ID	User story	Task	Time(h)
MA_ANDROID_1.1	As a guest I want to create an account.	create user register API	4
		create register UI	4
		develop register logic	4
		test user story	3
MA_ANDROID_1.2	As a guest I want to login from the application.	create user login API	4
		create login UI	4
		develop login logic	4
		test user story	3
MA_ANDROID_2.1	As a guest I want to consult all my reservations.	create getReservations API.	3
		create home UI	4

		develop logic	4
		test user story	4
MA_ANDROID_2.2	As a guest I want to filter reservations by status.	develop logic	3
		integrate filter button in the UI	3
		test user story	2
MA_ANDROID_2.3	As a guest I want to check reservation details.	create getReservationDetails API	5
		create details UI	4
		develop logic	4
		test user story	2
MA_ANDROID_2.4	As a guest I want to download amenity's documents.	create getDocs API	5
		Integrate buttons in the UI	3
		develop logic	4
		test user story	3
MA_ANDROID_2.5	As a guest I want to open the main building door.	create openMainDoor API	5
		Integrate button in the UI	4
		develop logic	3
		test user story	3
MA_ANDROID_2.6	As a guest I want to open/close the apartment's door.	create openListingDoor API	5
		create Reservation's details UI	4
		develop logic	4
		test user story	3
MA_ANDROID_2.7	As a guest I want to check/change the thermostat's temperature.	create getTmp API	5
		create changeTmp API	5
		Create UI components	4
		develop logic	4
		test user story	2
MA_ANDROID_2.8	As a guest I want to use Maps in	Integrate button in the UI	3
		develop logic	1

	order to get direction to the apartment.	test user story	1
MA_ANDROID_3.1	As a guest I want to send a message to the customer care service.	create sendMessage API	15
		Create chat UI	5
		develop logic	15
		test user story	4
MA_ANDROID_3.2	As a guest I want to check all my messages with the customer care service.	create getMessage API	15
		Create message UI	5
		create chat UI	5
		develop logic	5
		test user story	3
MA_ANDROID_3.3	As a guest, I want to get notification if I receive a message.	Implement push notification	8
		test user story	2
MA_ANDROID_4.1	As a guest, I want to browse all GuestWiser online deals using maps	create getDeals api	4
		Create Deals UI	4
		Maps integration	5
		develop logic	5
		test user story	2
MA_ANDROID_4.2	As a guest, I want to check listing's details.	create getDetail api	4
		Create detail UI	4
		develop logic	3
		test user story	2
MA_ANDROID_4.3	As a guest, I want to book an apartment using an online payment service.	Create setBooking API	5
		payment system integration	15
		develop logic	10
		Create payment UI	5
		test user story	4

Table 10 Sprint 3 backlog

Sprint 3 use case diagram

In this use case diagram, we will show all possible operations that the guest user can perform from the android application which are the same operations could be done in the iOS application except the invitation feature. We will have a global use case diagram alongside with four other sub diagrams that will explain these use cases: manage apartment's amenities and finally the customer care chat service.

This diagram is a global use case diagram in which we listed all the possible use cases.

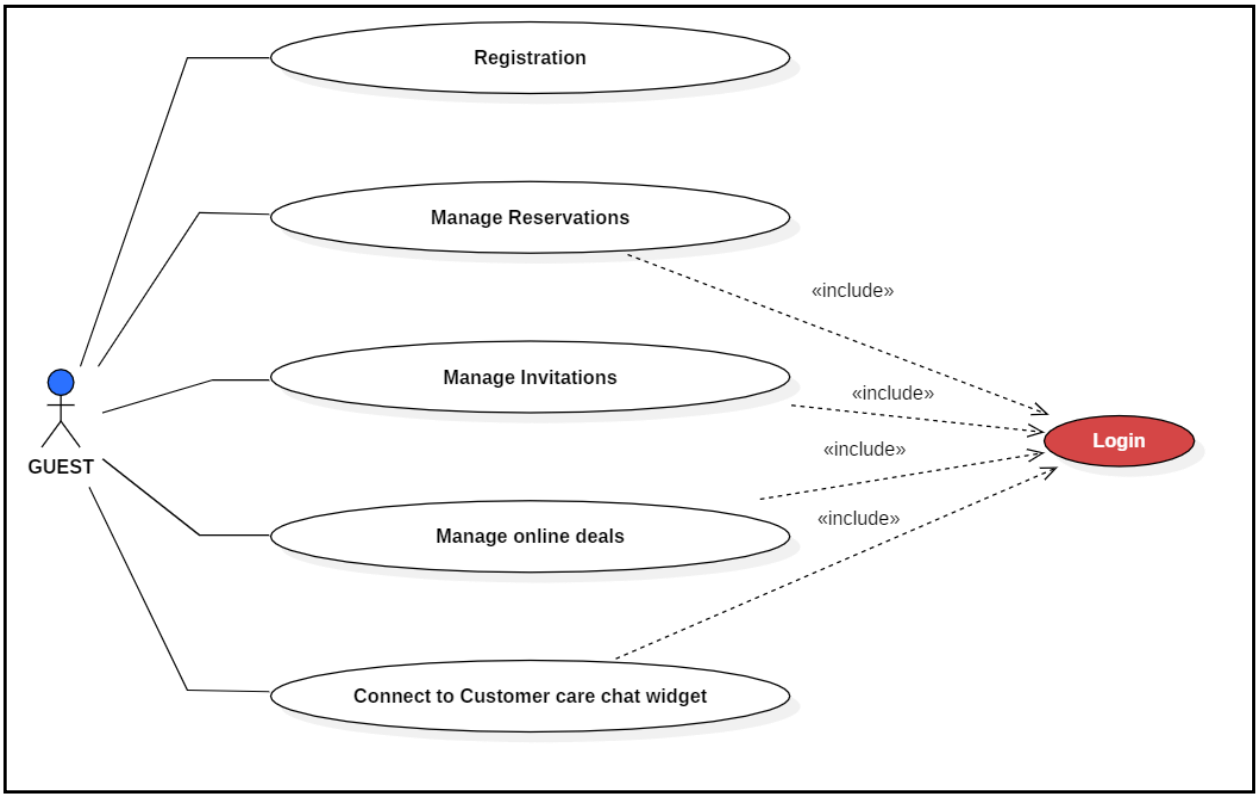


Figure 64 Sprint 3 use case diagram

The next diagram will explain the global use case "manage apartment's amenities" and divide it into smaller and more detailed use cases.

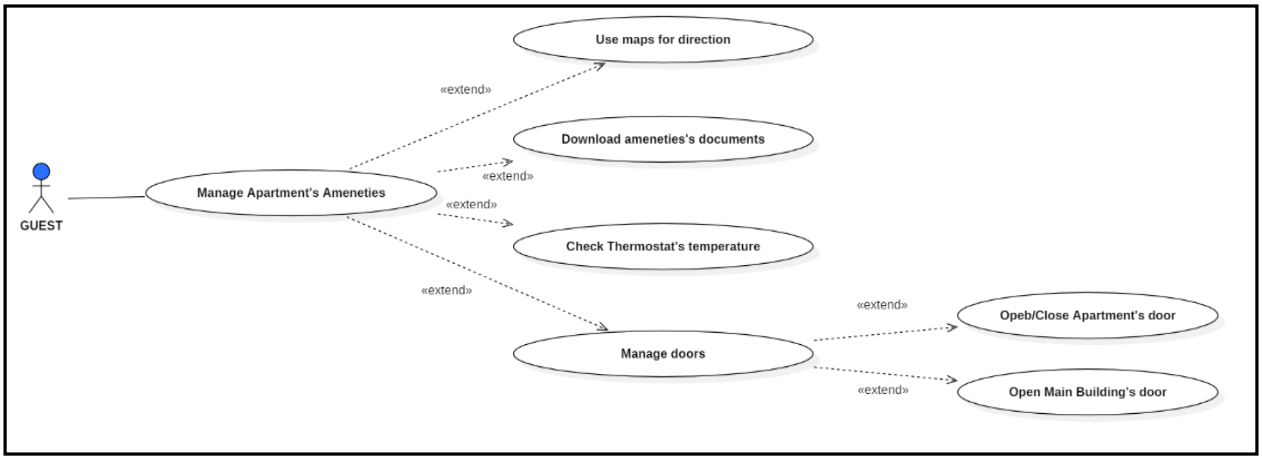


Figure 65 manage apartment's amenities use case

The second and final detailed diagram is explaining how to manage the customer care chat service.

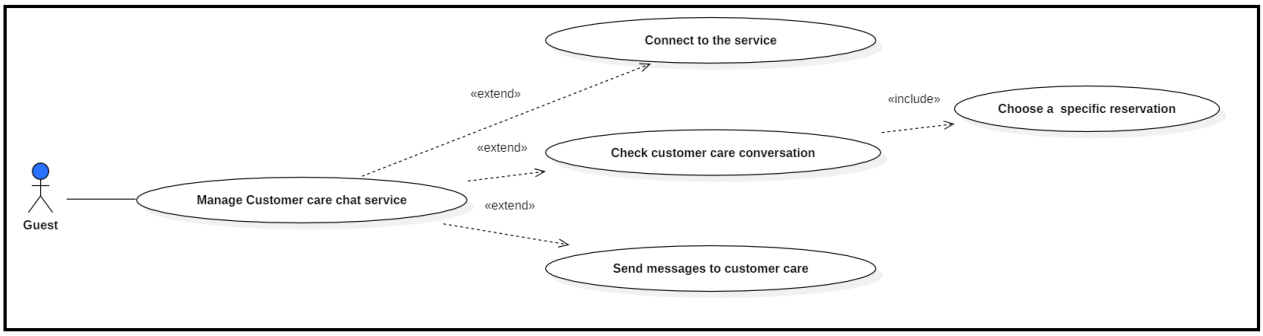


Figure 66 manage the customer care chat use

Use case 1: Open Main Building's Door

This use case is "**Open Main Building's Door"** which is a part of both Android and iOS applications where the client can manage to open and close the main building's door.

- **Textual description of "Open Main Building's Door"**
-

Use case	**Open Main Building's Door**
actors	Guest
preconditions	The admin should be logged in.
Post conditions	
Nominal scenario	1. The guest choose reservations from the menu.

	2. The system displays the list of reservations. 3. The guest click to check the details of his reservation. 4. The application displays the details of the reservation. 5. The guest click the button to open the main door. 6. The system perform a request to Point Central server in order to open the door for that reservation. 7. PoinCentral check if the user and the reservation are valid. 8. PointCentral notify the system that the door is opened. 9. The application notify the user that the door is opened.
Alternative scenario	• The application can't find any active reservation. • The application shows an empty list.

Table 11 Textual description of "Open Main Building's Door"

- **System sequence diagram**

This system sequence diagram explains all possible interactions between the guest and the web portal when he tries to check the active reservation reservations.

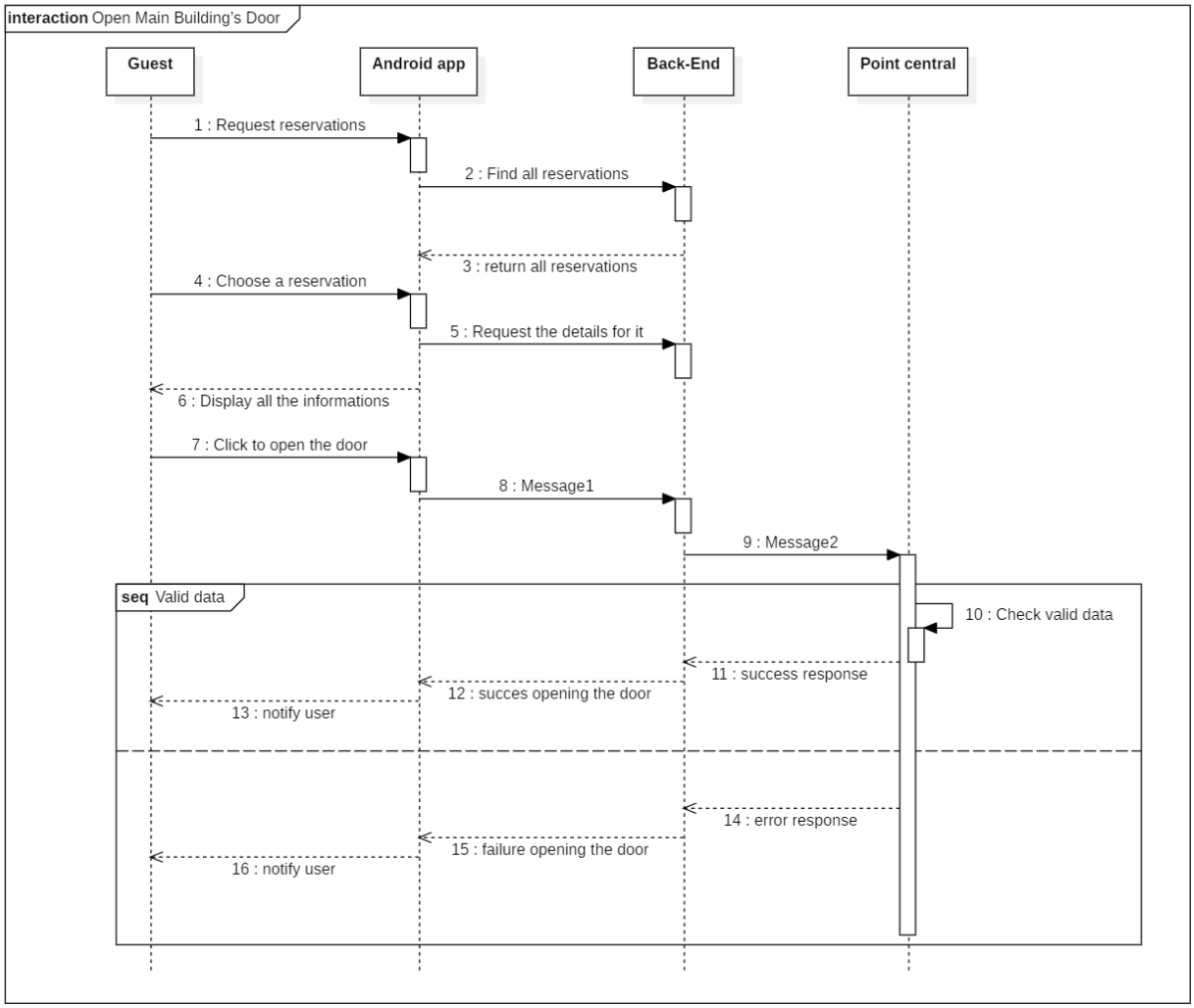


Figure 67 System sequence diagram of open main building's door

Analysis class diagram

The analysis class diagram will illustrate the classes of the iOS application. It will include all the new classes and the modification for the old ones in order to achieve our goal during this third sprint.

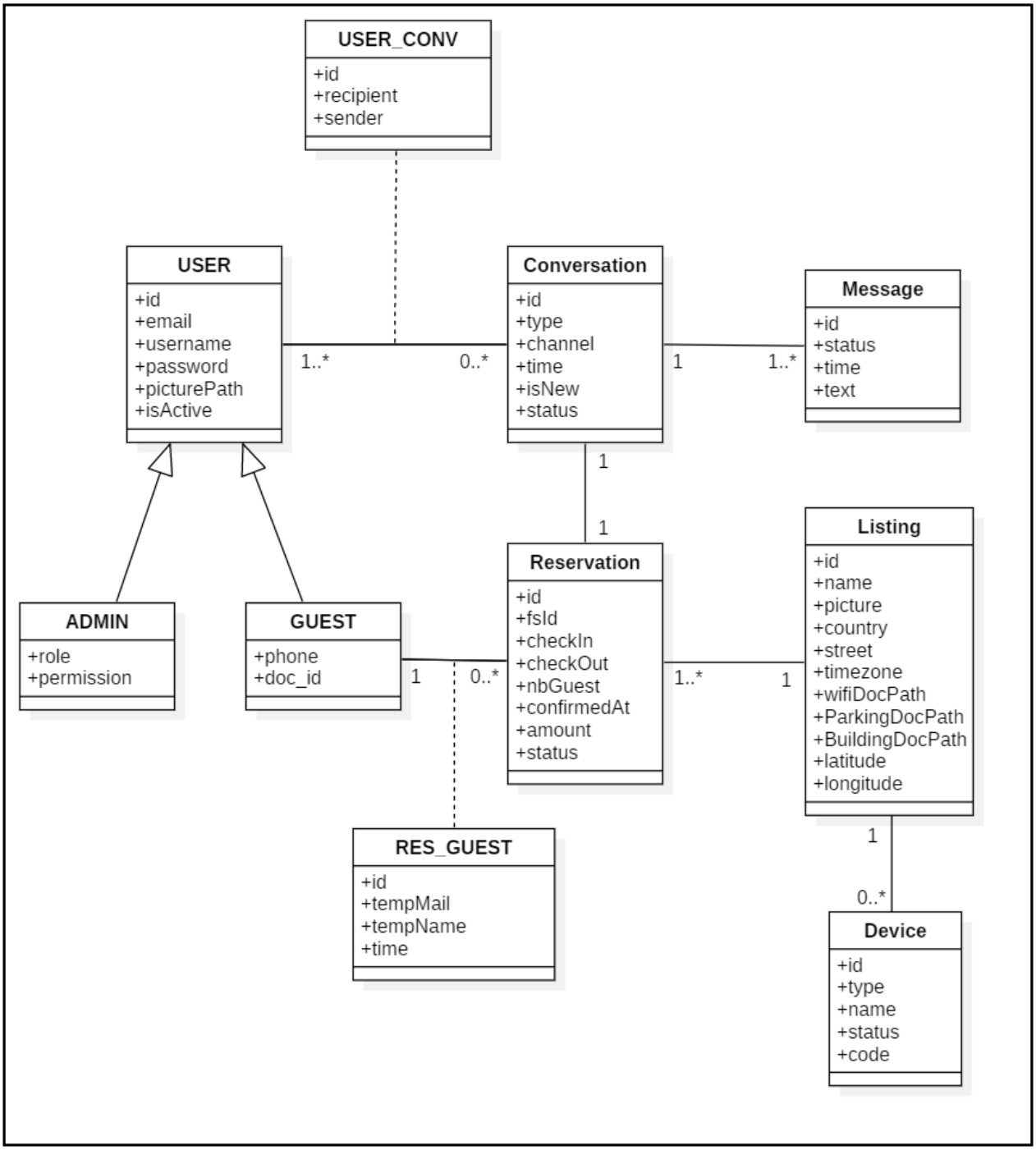


Figure 68 Sprint 3 Analysis class diagram

Sprint review

After performing the needed tests for each task in the Android application and finishing the development process, we made a demonstration and approved this sprint. The following screenshot is the register interface where the user fill all the needed information and also pick

a picture for his ID in order to create an account, a verification email will be sent to the guest in order to confirm his account.

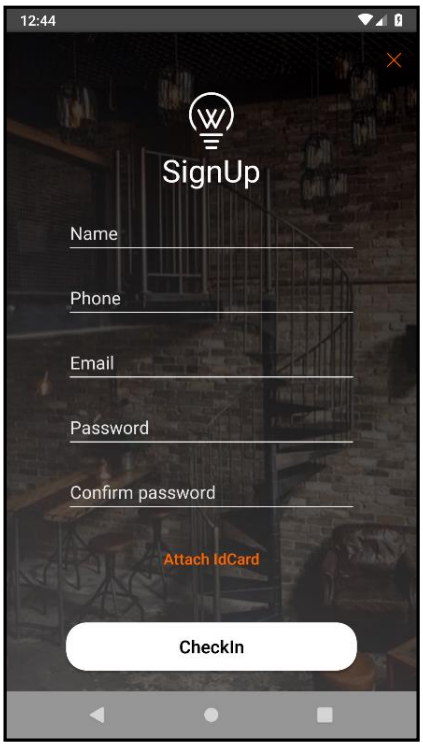


Figure 69 Registration interface

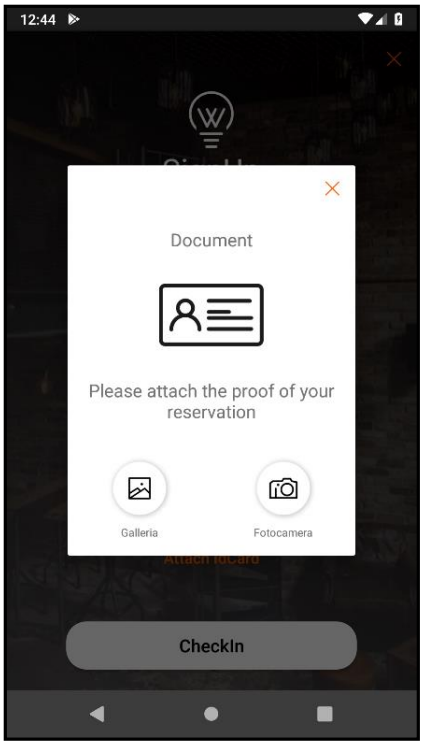


Figure 70 Upload id popup

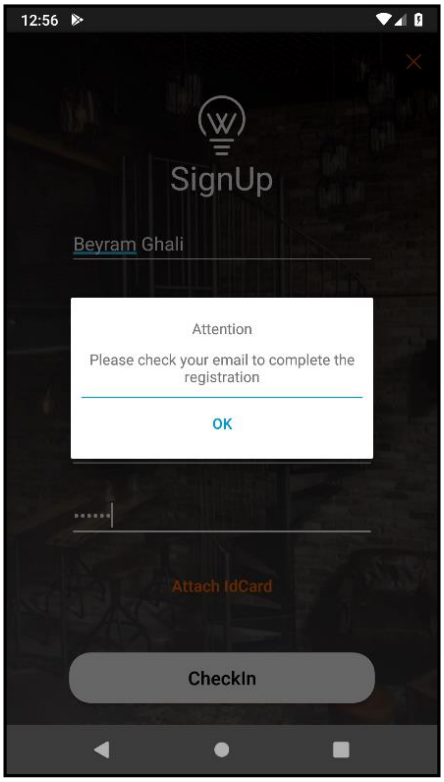


Figure 71 Registration confirmation dialog

- **Manage apartment's amenities.**

The next screenshot demonstrates how the user could manage his apartment's amenities by downloading Wi-Fi, building and parking documents, also how to recognize the direction for his apartment using Maps and finally how he could open/close the door automatically by choosing which action he wants from the dialog.

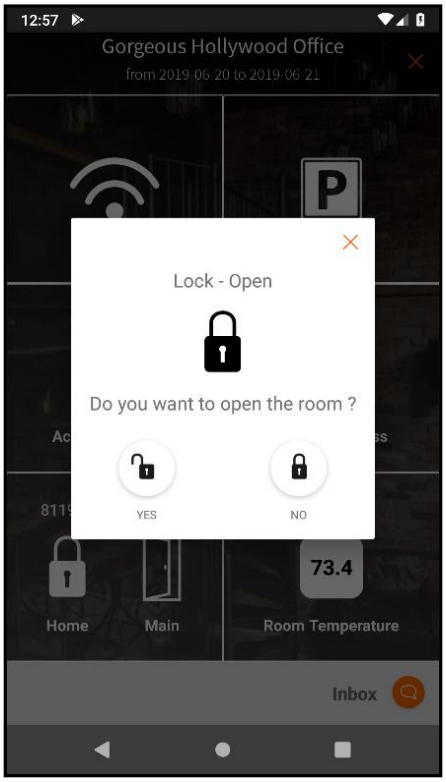


Figure 72 Open/close door popup

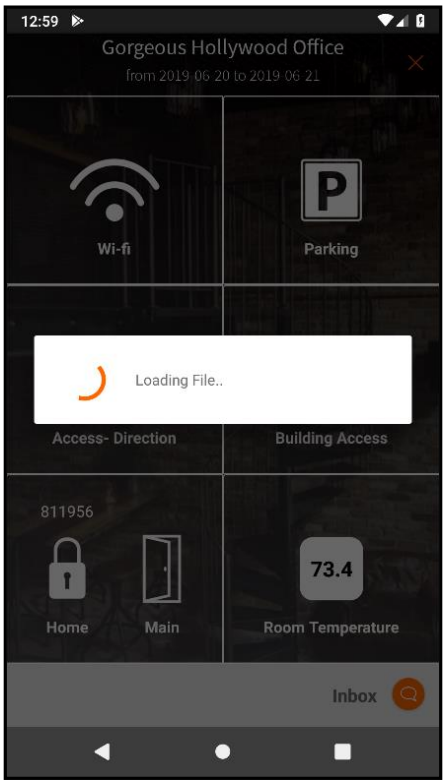


Figure 73 Download files popup

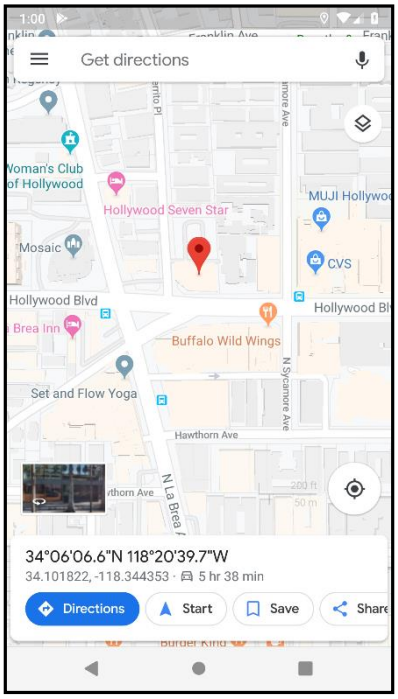


Figure 74 Map interface for apartment direction

- **Booking :**

The third section of this review show the booking part in the application which is still in progress due the delay for choosing the right payment system. But we used in this demonstration PayPal as a first payment method.

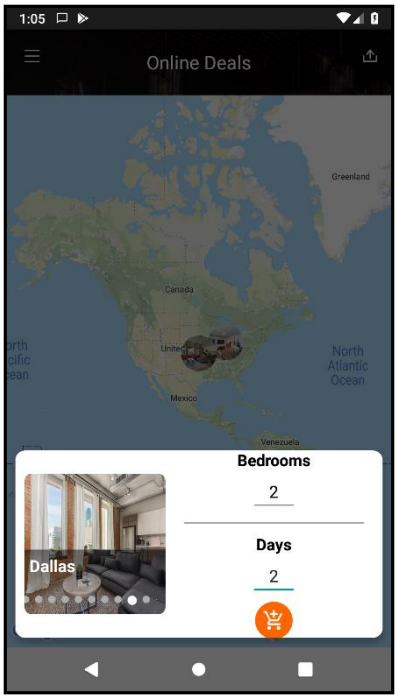


Figure 75 booking process interface

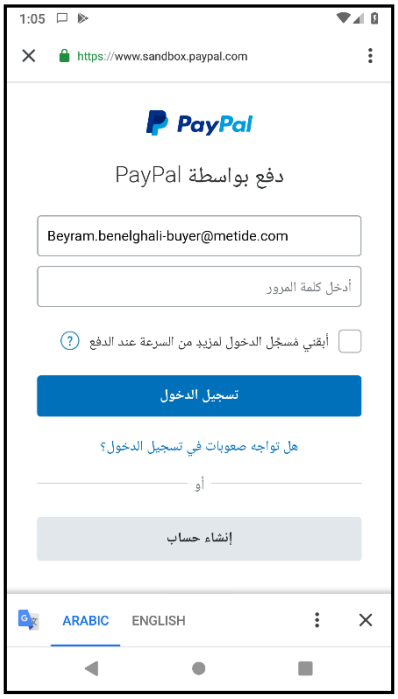


Figure 76 PayPal payment interface

- **Booking :**

The last section of this review show the integration of our customer care chat service which allow the guest to contact them in an easy way. The user will receive a notification every time he receive a message.

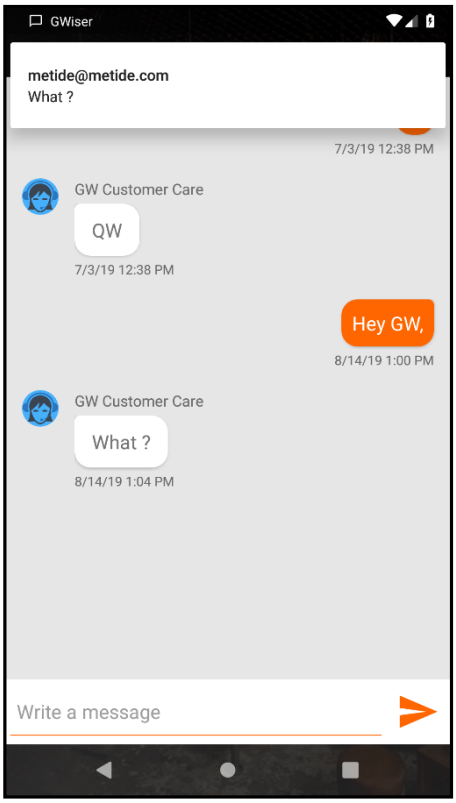


Figure 77 chat interface

Burn down chart

After finishing our tasks, we extracted the Sprint 3 burndown chart that illustrates the time spent in the development of the tasks regarding the expected timeline

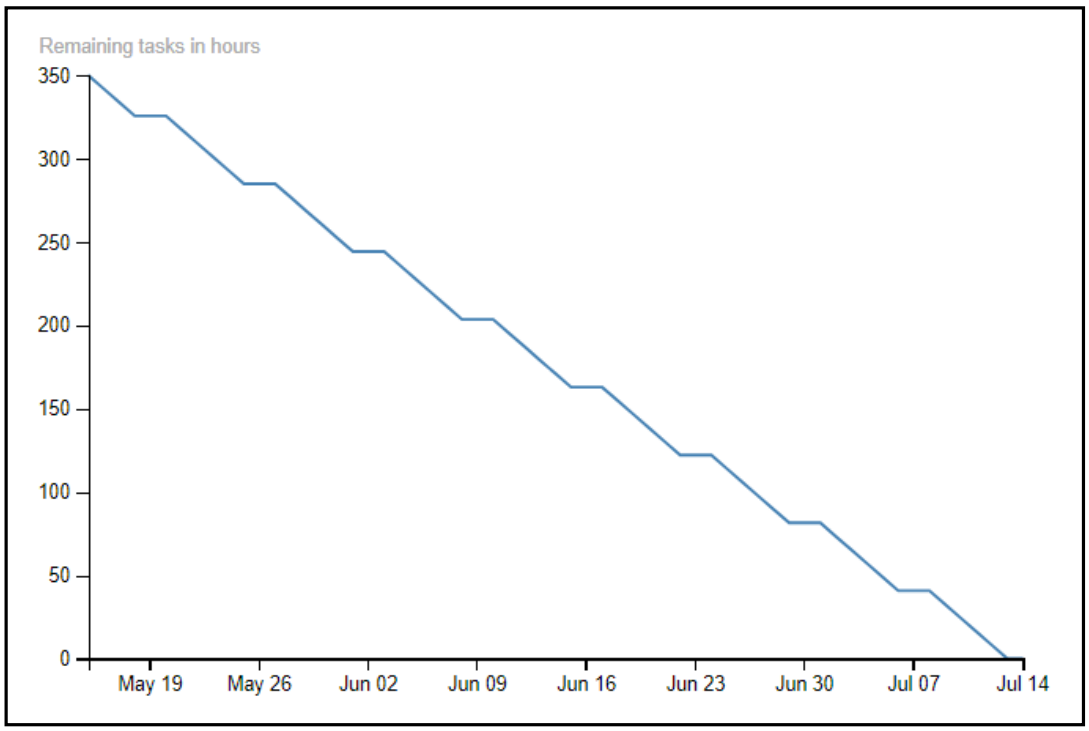


Figure 78 Sprint 3 Burn down chart

As a result of analyzing this chart, we can see that android application was delivered two days earlier than expected. It was submitted for review in the Google play store in order to allow the user to use it in their android devices.

Conclusion

In this chapter which presents the third sprint of the project, we have built, the android application for the guest and we have prepared all what customer need in in an efficient and

stable. In the next and the last part, we continue working on the next sprint for the development of the android version of our application.

Conclusion

This project targeted major problems that every real estate business owner faces which are usually solved separately but we gave a new integrated solution that will solve all of them, this solution is published as unified solution dedicated for guests and even for administration. Since we have a maximum level of availability in two different mobile platforms, we assured the satisfaction of all sorts of businesses and clients.

Our new project is still incomplete and need more improvements to be one of the best in its domain. For the next steps, we will work on adding new features on demand for guest, improving the design of the mobile applications also on increasing the security level by integrating one of CIA tools to identify criminals in order to make it possible to have a great experience outside your home.

Finally, this internship at Metide has been a great opportunity for me to improve my knowledge and have a various view regarding many things, both on professional and personal side, it helps me to develop my skills and learn how to speak with people in a professional setting which gathers different expertise and hierarchy levels and how to manage a project and handle stressful situations. I also got the opportunity to discover new technologies like Laravel which will allow me to build a mobile developer's profile who is ready to start his professional career.

Webography

- https://developer.android.com [viewed at July 2019]
- https://developer.apple.com [viewed at May 2019]
- https://angular.io/docs [viewed at February 2019]
- https://laravel.com/docs [viewed at February 2019]
- https://api.fsapp.io/docs [viewed at June 2019]
- https://stackoverflow.com [viewed at July 2019]
- https://www.pointcentral.com [viewed at February 2019]
- https://firebase.google.com [viewed at March 2019]
- https://skift.com/2016/06/17/the-business-of-hotels-vs-short-term-rentals-in-4-charts [viewed at August 2019]
- https://www.newgenapps.com/blog/what-is-agile-development-meaning-types-business-use [viewed at September 2019]

Urkund Analysis Result

Analysed Document:	REPORT_Ben Elghali Beyram (2).pdf
(D55299119) Submitted:	9/5/2019 3:10:00 PM
Submitted By:	imed.amri@esprit.tn
Significance:	2 %

Sources included in the report:

pfe report.pdf (D45600931)
https://skift.com/2016/06/17/the-business-of-hotels-vs-short-term-rentals-in-4-charts

Instances where selected sources appear:

3

ESPRIT SCHOOL OF ENGINEERING

www.esprit.tn - E-mail : contact@esprit.tn

Siége Social : 18 rue de l'Usine - Chargula II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : Z.I. Chotrana II - B.P. 160 - 2083 - Pôle Technologique - El Ghazala - Tél. : +216 70 685 685 - Fax. : +216 70 685 454