Frankfurt University of Applied Sciences
Masters of Engineering
Information Technology

Course:

Agile Development in Cloud Computing Environment
Dr.Patrick Wacht
Faculty 2: Computer Science and Engineering

Project Title: Accommodation Service in Travel App

Authors:

Maria Ranjbar   1324772   maria.ranjbar@stud.fra-uas.de
Junaid Khalid   1364677   junaid.khalid@stud.fra-uas.de

Summer 2022/2023

Table of Content

Abstract

In today's world, we know how technology plays an essential role in our life, and we cannot imagine ourselves living without it. This paper aims to provide an online accommodation service as part of services offered in Travel App. We tried to use a simple method to create a platform that is easy to use by user and at the same time will fulfill the user's expectation.

We looked into every aspect of this service to create a platform which will not confuse the user when opens it. That means, it is important how convenient the app should be as well as being able to use it sufficiently. In addition, we needed to pay close attention to our users' needs and provide the service which can meet their expectations.

We started doing research about what our user demands most on accommodation service, after that we provided features regarding those demands in our Kanban board, then we defined our user story, context scenario as well as key path scenario which we believed it could give us a better sight to move forward, finally we used Figma software to draw our user flow and wondershare Mockitt software for designing our prototype. The last step, we have taken was to use python programming language along with SQLAlchemy data base table for building the logic and data handling. We used the bases of Flask 1.1.2 in order to achieve our API functionalities.

1. Introduction to Agile Environment & Scrum

An agile environment is described as one that fosters and promotes a culture that motivates a team of individuals to work together toward a common objective. This is accomplished through infusing the significance and value of people and their interactions into corporate culture, particularly in terms of working to attain excellence, teamwork, and acceptance of frequent change.

Since approximately a decade ago, the term "agile" has been popular in project management, and with good reason. Agile is essentially a catch-all word for a number of project management techniques that are distinguished by their capacity to let project teams adapt to shifting priorities and requirements through incremental work packages. Despite the fact that all agile approaches have some traits, they all have distinctive processes that make them stand out. Let's take a closer look at how each technique is applied by the team.

1.1. How Scrum Works?

A framework called Scrum is used to create and maintain complicated products. Let's see how a scrum operates. We discuss the product backlog with the client as our team starts a sprint, an iterative cycle. In essence, the customer's prioritized feature wish list is what makes up the product backlog. Within a two- to four-week timeframe, we pick the most important activities to work on for that sprint. The sprint backlog is the name of this task list.

The team holds a brief meeting each day throughout the sprint called the daily scrum to decide what work needs to be done and to address any delays or obstacles. The team meets with the client to deliver the finished product, go through the product backlog, and reorder the priorities the specifications. Up until the complete project is built, the team keeps working in sprints.

1.2. Kanban board

Using an agile management technique like Kanban is a terrific approach to map the value stream of work process and produce a more adaptable workflow.

The team becomes more adaptable to a changing environment. Customer communication also becomes more transparent as the job is visualized. Customers can know what you're going to do next by using tools like shared boards, backlogs, and Kanban cards, for instance. This enables you to immediately react to any necessary changes in requirements that may occur throughout the development cycle and to readily convey the definition of done with them.

You can use it to improve your workflow's efficiency and agility by:

➢ Transparency
➢ Adaptability
➢ Enhanced Team Cooperation

Agile Development in Cloud Computing Environment

 We could see what each of us was working on because the Kanban board displayed all tasks and their current state. This helps to clear up the mess, puts us all on the same page, and eventually speeds up how project moves through our work system.

We started small and kept the mapping process simple without overcomplicating things. In the beginning, we divided our workflow into the three basic stages: "To Do" (work that needs to be done), "In Progress" (work is in motion),"Review"(any changes need to be deployed) and "Done" (work is completed).

In addition, we defined nine sprints for our workflow with deadlines, so this way we could commit our tasks (push &pull) daily and prevent missing the deadline. In each sprint we made an effort to include summaries of what should occur during each work phase. In this way, everyone on our team would be aware of the particulars of each phase in the process.We had our weekly meetings in order to discuss the process of our workflow and how to make it even better ,we were able to exchange our opinions on workflow to improve our project and by this we could find a solution if we would encounter any complicity.
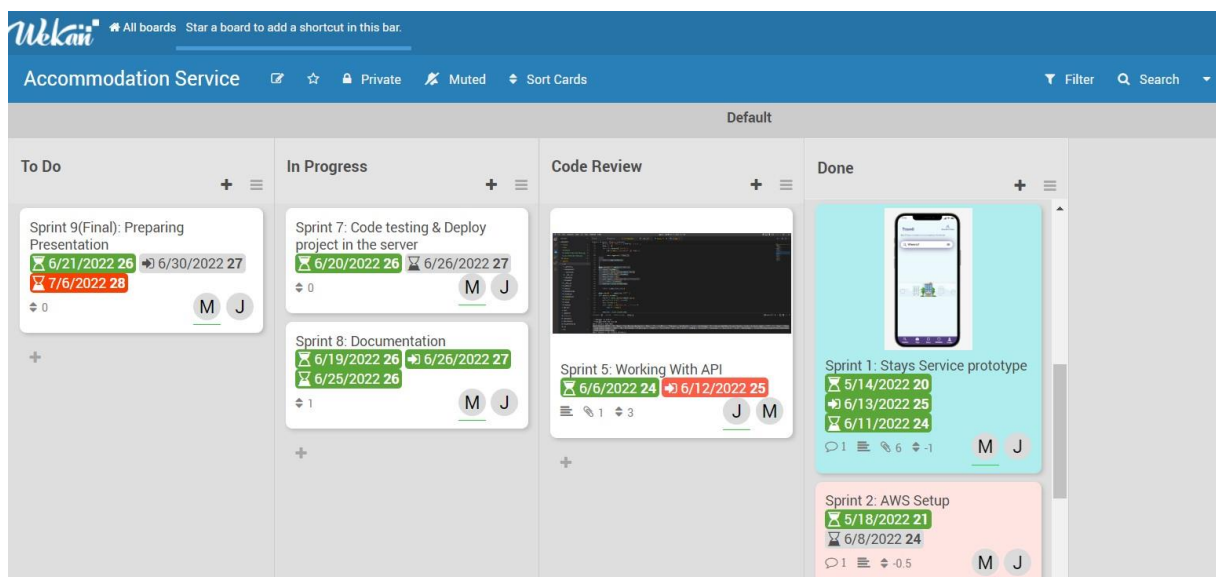

Figure1.Kanban Board

By working on Kanban board, we realized that we do not have to stuck with the task we do not know much about it, instead we tried to take small steps and get done the tasks we already have the knowledge, and later on we gathered info about complicity of that specific card and tried to move forward, In this way we would not waste time and our deadline.

Based on our data, we assigned a particular likelihood. Saying that a feature will be completed with an 85% probability and in the next 5 to 8 days, for instance, is considerably more feasible than giving an exact completion date, which is agile planning practice technique.

One interesting idea of using Kanban board for us was, how it makes the team cooperation easy and practical as well as saving time which was a big point for us.

2. Steps Taken in Design process

2.1. User Persona

While we were working on accommodation service as part of travel app product, we knew that we need a user persona, so we finally came to conclusion to create one. In our user story, Maria Smith is a student who likes traveling

and exploring the world. What has been always an issue for her while she was traveling was, she has to spend so much time to find an accommodation and this is really frustrating for her going through different apps and websites, and not to be able to book something easily. What she encounters all the time is the complexity of these apps and she would like to use an app which is simple and user friendly at the same time.

2.2. Context Scenario

Maria is planning to travel to Barcelona, Spain this summer during school break. She already booked her flight ticket but unfortunately, she could not find the right accommodation yet to book. On the other hand, she does not want to spend so much time on the websites to find a place to stay, she prefers to find a room in a hotel where she can enjoy her adventure.

2.3. Key Path Scenario

Maria finds our accommodation app; she chooses the destination city Barcelona and goes through filter option. She can select the date, number of rooms as well as price range. Our app will show a list of hotels with the specific needs she has chosen. She picks one hotel and sees the facility that has been offered such as breakfast, parking, and reads the review then she decides to request a booking. She follows up with the booking form and fills her personal information, and finally books the hotel room she wanted.

2.4. User Flow

Based on our key path scenario we created our user flow in order to show a better understanding of what we are going to create as our product.

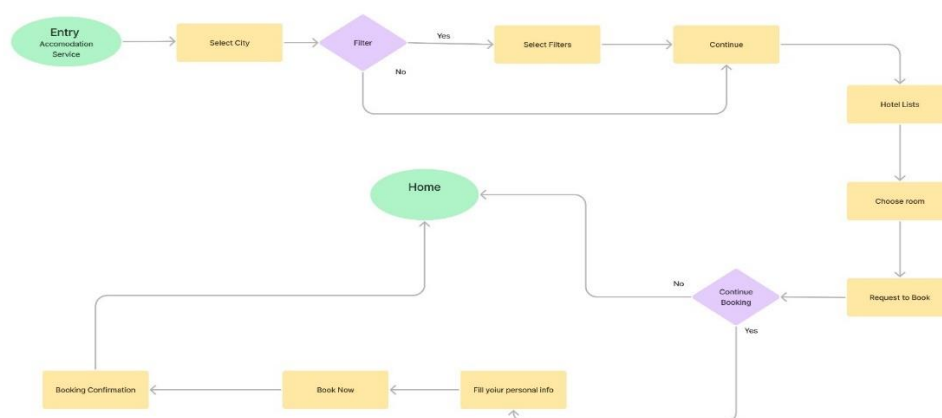Below you will see our user flow diagram:

Figure 1.1-User Flow

2.5. Prototype

We thought the best way to simplify what exactly our service will offer to our user is to design a prototype based on our user persona and user flow, before coding. As the customer will get an idea that how this service will work at the final phase.

Here's the link to our prototype:

https://mockittapp.wondershare.com/app/l4vqnuo9mauctp?simulator_type=device&sticky

3. Implementation of code

3.1. Tools and technology:

The tools and technology that we have used for creating this project is visual studio code as IDE with Flask 1.1.2 for API development. On the other hand, Flask is a minimal framework that just offers the essential tools, extending its capability through the use of other libraries and frameworks. You choose just what you need for your project, nothing more, which is fantastic. the programing language which we have used is python 3.7.4 along with the data base SQLAlchemy. The detail libraries and their dependence are shown in the table.1

Package	Version
aniso8601	8.0.0
certifi	2022.6.15
charset-normalizer	2.0.12
click	7.1.2
Flask	1.1.2
Flask-RESTful	0.3.9
Flask-SQLAlchemy	2.4.3
idna	3.3
itsdangerous	1.1.0
Jinja2	2.11.2
MarkupSafe	1.1.1
pytz	2020.1
requests	2.28.0
six	1.15.0
SQLAlchemy	1.3.18
urllib3	1.26.9
Werkzeug	1.0.1

Table. 1

For the request purposes to our created APIs, we are using Postman software.

3.2. Database Model:

Data base model which we have consider for this project is shown in the fig-1.2. We used this database model for the during the planning of our project because of the following benefits.

it gives us a highly efficient method for handling large number of different types of data with ease.

• It allows large amount of data to be stored systematically and these data to be easily retrieved, filtered, sorted and updated efficiently and accurately.

The type of the data base model for this project we selected is relational database model. This model uses a system to maintain the relations using RDBMS (relational database management system). The data is arranged in tables of

different entities with a primary key. There are three main components of this model which are relations, attributes and domains. Every column starts with a name which is called attribute and row are knowns as tuple. Cardinality of any relation is defined by the number of tuples. The degree of relation is determined by the number of attributes a table is having. There are two types of keys used to identify a table Primary key and foreign key. Primary key is a unique id of the table and if you export your primary key PK to another table is becomes foreign key FK. You can see them in the fig-1.2
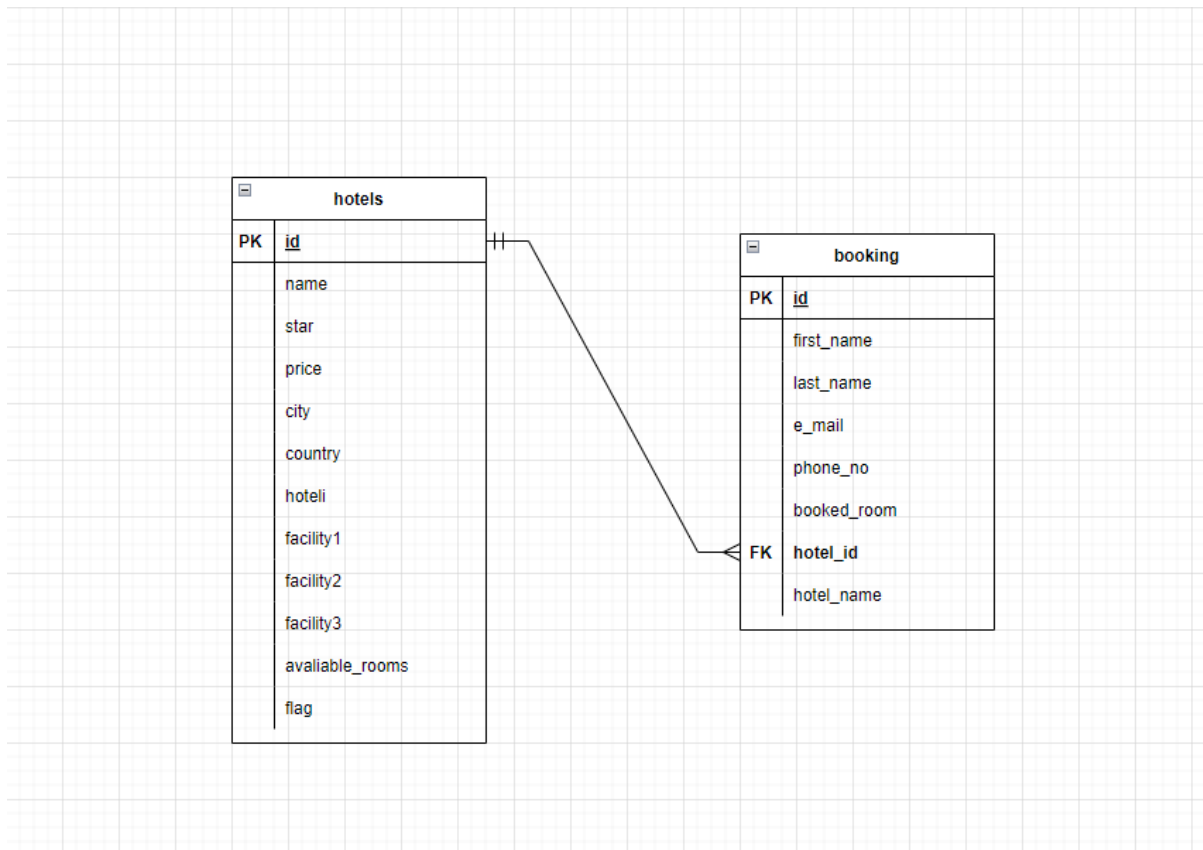

Figure-1.2

In fig-1.2 In the database, there are two tables that provide information on advertisements for accommodations. Every record's "id," which is comprised of the hotel's or accommodation's name, has a unique identification number. "Stars" is an integer value that only serves to reflect the accommodation's rating because it is an important factor for consumers who want to make a reservation. The cost of the hotel is indicated with the word "price." The two factors that best describe the location of the hotel are city and country.

For the purposes of example, hoteli stands for the hotel image url; however, many images can be added by using the format hoteli1, hoteli2, etc. Additional indulgences that a consumer might purchase if they book lodging are facilities 1, 2, and 3. "Available rooms" is an integer value that displays the rooms that are available for reservation in the lodging. One room may accommodate two adults and has two beds and a bathroom. We utilized the filtering variable "flag" to distinguish between hotels with full bookings and those with available rooms. Due to their fullness, those hotels that have no vacant rooms will not appear in the hotel availability API.

7

All a hotel's bookings are listed in the "booking" table. This often includes the customer's contact information and personal information so the hotel personnel may get in touch with them about their booking request. Additionally, this table has a distinct ID. The customer's name is contained in the fields marked "first name & last name." "email & phone no" are the customer's contact information. The number of rooms that have been requested for booking is displayed by "book room". The hotel's unique identifier, "hotel id," which is imported from the table of hotels, is what you use to make reservations there. You can see that the relationship ship line indicates that many reservations for the same hotel are allowed, and that the hotel name will appear based on the hotel id as the last option. When these details are combined, a room reservation will result.

This database model can be expanded based on what new features you want to incorporate i.e., adding a customer table for holding records of login details of the customer, add a payment table which will have bank details for

payment purposes. But as for our requirement in the available time frame we are able to achieve our 3 purposed APIs using this database.

3.3. API Planning:

During our planning phase as discussed above we decided to have three API end points:

1. Stays
2. Become a host
3. Booking

3.4. Stays:

The API for this is:

http://127.0.0.1:5000/hotels/

You must submit your request using the "GET" method. It will list every hotel in the database that is open for reservations. Hotels that do not have any available rooms will not appear in this API. This API's output will be in .json format, as demonstrated below:

```
[
  {
   "id": 0,
   "name": "Les Jardins du Marais",
   "star": 7,
   "price": 400,
   "city": "Barcelona",
   "country": "Spain",
   "hoteli": "D:/Lectures/AgileRepo/Project/tmp/Les Jardins du Marais.jpg",
   "facility1": "Parking",
   "facility2": "Pool",
   "facility3": "Heating",
   "avaliable_rooms": 10,
   "flag": 1
},
```

```
  {
     "id": 1,
     "name": "Golden Tulip Little Palace",
     "star": 7,
     "price": 410,
     "city": "Barcelona",
     "country": "Spain",
     "hoteli": "D:/Lectures/AgileRepo/Project/tmp/Golden Tulip Little Palace.jpg",
     "facility1": "Parking",
     "facility2": "Pool",
     "facility3": "None",
     "avaliable_rooms": 10,

     "flag": 1
  }
]
```

As shown, there are two hotels added in the database and you can see all other attributes which are describing the hotels. All the hotels available for booking will show up using this API.

3.5. Become a Host:

<p align="center">http://127.0.0.1:5000/hotels/<int:hotelID></p>

this URL which is adding the new hotels for the booking.

The "PUT" method must be used to add hotels to the database. As indicated in the URL, you must provide a unique id, i.e., /hotels/0, and then you can send the other parameters in the body to add it to the list of hotels that are currently available. Name, star, price, city, country, hotel, and available rooms are a few of the body's fields that must be filled in. However, adding facilities is optional; a maximum of three facilities may be added. The figure-1.3 shows how to include a hotel. The table "hotels" will include all of the entries.
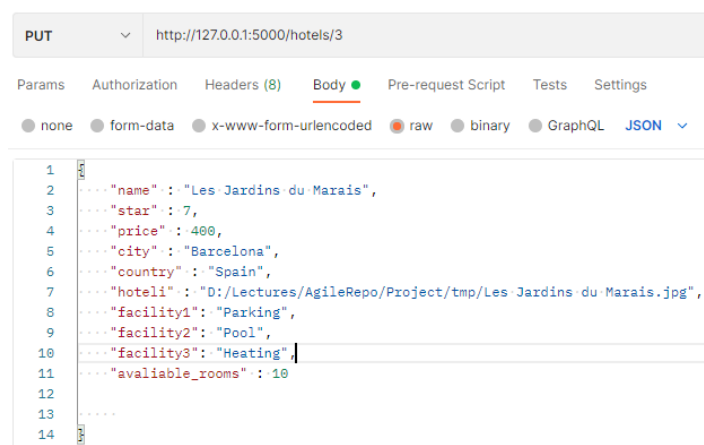

Figure-1.3

Additionally, you may use the "GET" method to obtain a particular hotel and the "DELETE" method to remove a hotel. You don't need to add anything to the body for these two techniques. It will function simply with the <int:hoteID> provided.

3.6. Booking:

http://127.0.0.1:5000/hotelbooking/Les Jardins du Marais

You must use the "PUT" method to make a reservation here. Instead of supplying ID, you will now need to submit the name of the hotel where you want to make a reservation. Your reservation's integer ID will be generated independently and will never be used twice. Until all of the rooms are reserved at a hotel, reservations may be made at multiple times. Once you enter the hotel's name, it must appear in the list of possibilities; otherwise, an error will be generated. You will be able to finish your reservation if the hotel is there. The following parameters must be specified in the body of your request before you submit it, as shown below:

```
{
   "f_name": "Malin",
   "l_name": "Vonhagen",
   "e_mail": "malin@hotelbooking.com",
   "ph_no": 015737944234,
   "broom": 1
}
```

Once you have booked your request you can also see all the booking that have been done one every hotel using "GET" method and by providing the name of the hotel. Reservation in hotel "les Jardins du Marais" is shown in the figure-1.4:
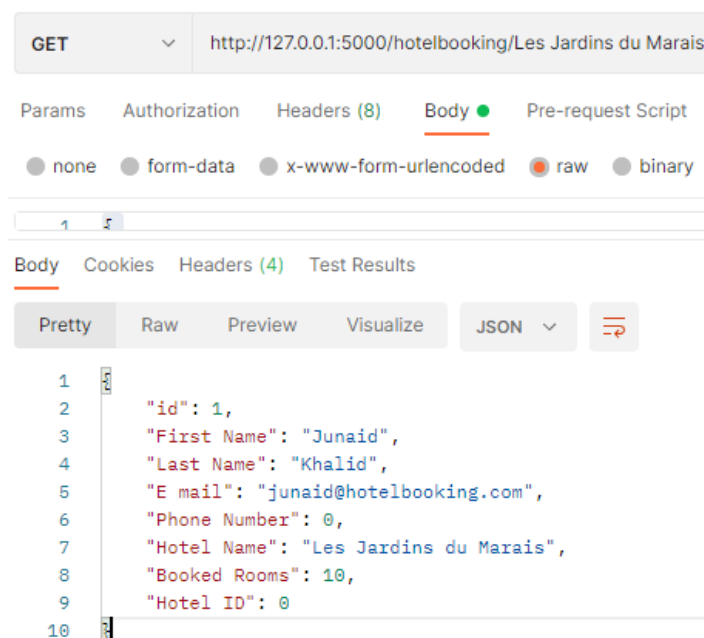

Figure-1.4

This figure illustrates a "GET" request, and the response is displayed below with the "id:1" signifying that your booking has been completed. If there is an entry for the hotel you want to reserve a room in, the Hotel ID will be a foreign key that will take from the other table "hotels" in the database. These reservations are being kept in the database's

"booking" table. Using the same API, you can also erase all hotel reservations by using the "DELETE" method. You only need to give the hotel name in the API for deletion.

The code may eventually include new APIs to add more functionality to improve services for the lodging industry. We were only able to offer the three APIs we had intended owing to the short time schedule.

4. Conclusion

As the global technology innovation is growing fast every day, the more you know about your user as a target group the more you can produce a product for that category. We think we could understand what our user needs when going through our accommodation service, therefore we planned to have an accurate data regarding the needs of our user, as well as attractive product design, and performance simultaneously. There are still capabilities open to this service which can be expanded time by time and we hope we can work through that in the future.

5. Resources

1. https://docs.microsoft.com/en-us/devops/plan/what-is-agile
2. https://careerfoundry.com/en/blog/ux-design/the-ux-design-process-an-actionable-guide-to-your-first-job-in-ux/
3. https://www.interaction-design.org/literature/article/how-to-visualize-your-qualitative-user-research-results-for-maximum-impact