University of Puerto Rico
Mayagüez Campus
Department of Computer Science and Engineering

Gym Schedule Application: Proposal

Group 5, March 6th, 2022

Group 5

Andre A Clavell Alvarez
María H Cotto Nieves
Jose L Lazcano-Etchebarne
KEVIN R LINERA-HERNANDEZ
Fernando Ramos Lugo

Grace M Fernandez Rivera
Gabriel Garcia
Alaina N. Del Moral Martínez
Sebastian O Espinosa
Luis Adorno
Sonia Ortega Espín

I. Informative Part

1. Name, Place, Date

Condominium Gym Schedule, Mayaguez, PR
March 06, 2022

2. Current Situation

Since the pandemic started people have been using their local condominium gym more often than commercial ones due to the high risk of getting infected from the virus. In a Morning Consult poll, only 20% of Americans are comfortable going to the gym, and 25% of Americans don't plan to go back to commercial gym's as of July 2021. Most condominium gym's still have some regulations in order to keep their residents safe during the pandemic. The most common rules are the limited capacity of people using the facility at the same time, and the amount of time you can stay on the facility. Residents don't have a way of knowing if the gym is already on maximum capacity until they get there. This causes a problem to the residents since they already allocated the time to exercise but must now wait for someone to finish in order to enter the facility. This results in the resident having to wait outside, or simply leaving.

3. Needs

For the condominium gym goers there is a need to have control over the time they are going to use the facility. This need includes knowing the availability to separate the gym for periods of time each day. There is also a need for the gym goer to cancel their booking if they can't go anymore, this gives other people the slot if they were interested in going at that given time.

4. Ideas

We will develop a time scheduling application for a condominium gym. This application will contain a calendar with available time frames for each day, where residents can separate specific times to use the gym facilities. Additionally, the application is going to have a feature where users could notify the next user in line to use the facility if they finished early, intended to have better efficiency of time. To ensure that a single person does not schedule multiple time frames in a single day, the application will only allow the reservation of one time frame per day. There will be a specific hour of the day where users can schedule the available time frames intending to give the users a "fair shot" to schedule the time they want.

5. Scope & Span

For the scope of the application would be the collection of residents using the gym session app. Our system would develop to have control of users and proof authentication as residents of this condominium. Is a system that expects to be very favorable for both parties, administrative and for residents. Time-management is a key for residents and the space-management for the gym administration.

Span for this application be rated by residents. The administrative account can activate the users accounts with the resident number and full name that work as identification cards and authenticate their registration. Each user is limited by a 12 hour schedule/ cancellation period. Let a single user separate up to two spaces for gym and with a restraint only for 1 hour time slot limitation. The Gym space has a limited number of people inside the gym that would be monitored by the system.

6. Synopsis

We will develop an application that functions as a way to schedule certain time frames in the condominium's gym. This application will allow users to book their available or preferable time to exercise and as such they do not waste time verifying if there is any space available in the gym or not. The user will be able to see when the gym is available, and may schedule in advance. A notification system alerting the user that the space they scheduled is available, or if for any pertinent reason it has been canceled should be included. To implement this system, on the frontend we have the choice of React or Angular. As for the backend, some tools available are Django or Flask. Finally as for testing, the options available are Pytest and cucumber.

7. Partners

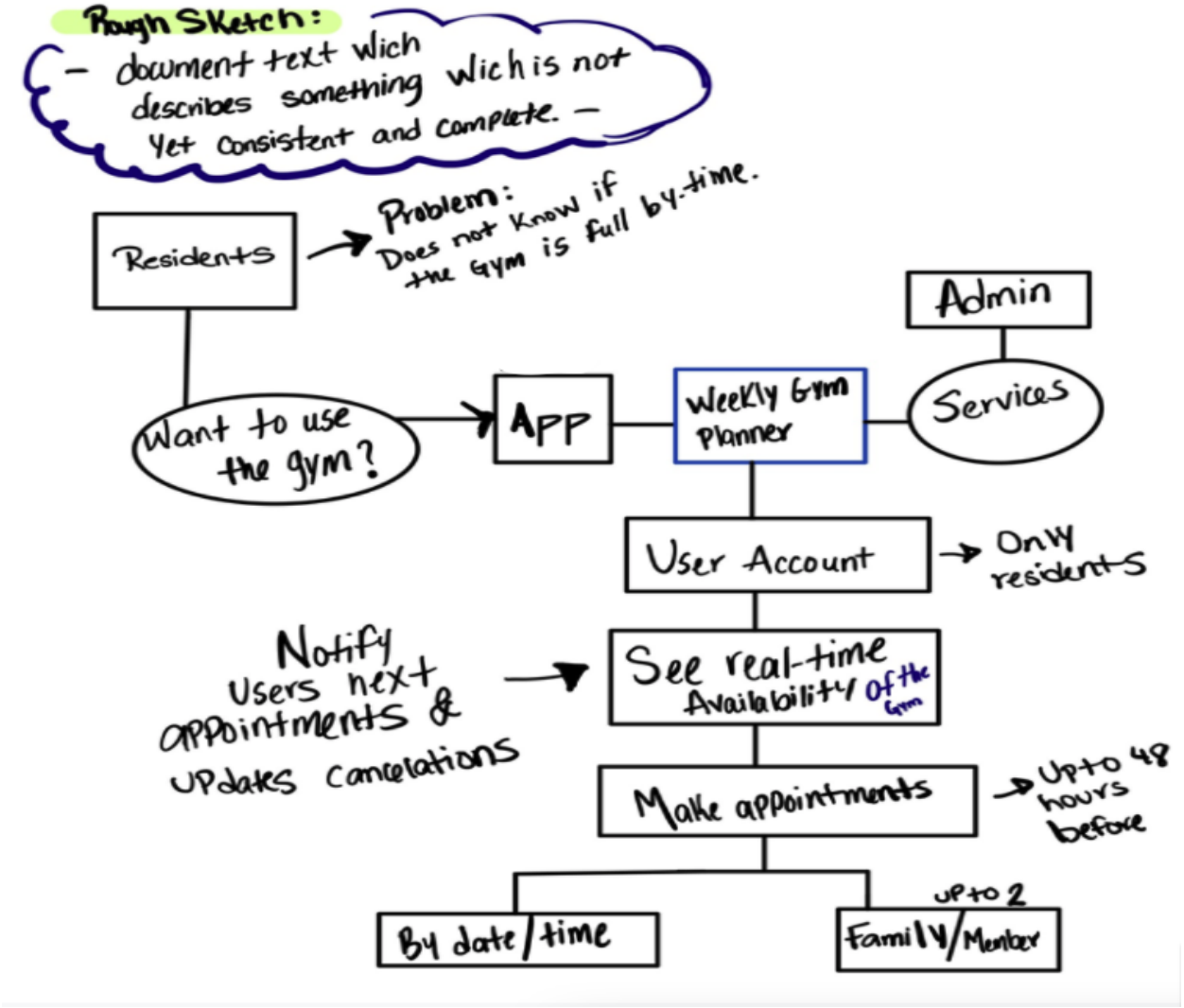
The partners of the project are developers, the condominium administration and its residents. The condominium administration will be able to oversee the amount of residents using the gym at any given time as well as restrict entry for maintenance purposes etc. Residents will be able to choose any given time slot that is available to be granted access to the gym.

8. Assumptions

We assume that the development team has sufficient knowledge of the requirements and domain of the project. Additionally, the schedules requested by the users will be correctly slotted into the time that they chose avoiding overlap when the user limit for the gym is met. We also assume that the schedules will be accurately displayed for all users to avoid confusion at the time of requesting a time slot. Also we will not include information on who is scheduling at whatever time to avoid conflicts.

II. Descriptive Part

1. Rough sketch domain description



In the image above, we can view a rough idea of what we had planned for the application. This describes the process of the user separating a slot of time at a given week.

2. Domain Narrative

A resident is going to the gym and is limited on time. They could go to the gym hoping for the chance that the resident limit has not yet been reached but this could lead to them simply wasting time and not knowing

how much longer the other residents are going to take in the gym in order for them to be allowed in to do their routine. They could come now and again to check for availability but in the end the time wasted would probably not make the effort worthwhile. Our solution to this problem is to create an application where residents can reserve a certain time slot that is available to them while meeting other requirements such as the resident limit making the time wasted in checking for availability a thing of the past allowing residents to be more efficient with their time.

3. Rough sketch requirements as stories

- “I haven’t been able to find a moment where I could go into the gym, everytime I come down to check it's already filled up again. If this goes on my only workout would probably just be going up and down the stairs hoping that the gym is available.”
- “Every time I get ready to workout, when I arrive at the gym facility it's always full and it's a waste of time."
- “I wish there was an app that I can reserve time to use the gym and organize my time more efficient”

4. Requirements

- **Interface Requirements:**
  - **REQ-INT-01:** The system shall have a register page, where residents can create an account.
  - **REQ-INT-02:** The system shall have a login page, where users can log in to their accounts.
  - **REQ-INT-03:** The system shall display the login and register page at the top of the home page. The register page will contain five blank spaces for the Full Name, Email address, apartment number, password and password confirmation. The login page will contain two blank spaces for email and password, respectively.

    - **REQ-INT-04:** The system shall let users know when their account has been created at the top of the homepage.
    - **REQ-INT-05:** The system shall display the “Gym Schedule” option page on the top of the home page.
    - **REQ-INT-06:** The system shall display the available time frames that the facility can be used on the “Gym Schedule” page.
    - **REQ-INT-07:** The system shall display the active reservations that the user has.

- **Domain Requirements:**
    - **REQ-DM-01:** The system-to-be shall allow the user to schedule a specific time frame of the day.
    - **REQ-DM-02:** The system-to-be shall not allow users to schedule multiple time frames of the day.
    - **REQ-DM-03:** The system-to-be shall allow the user to change or cancel the scheduled time.
    - **REQ-DM-03:** The system-to-be shall allow the user to notify if they finished before the time scheduled, and send a notification to the next user in line.
    - **REQ-DM-04:** The system-to-be shall order the available time frames in chronological order.

- **Machine Requirements:**
    - **REQ-MACH-01:** The system shall have a database where all the information and reservations of the users is stored.
    - **REQ-MACH-02:** The system shall encrypt/decrypt the user’s information before storing it on the database after registering or logging in.
    - **REQ-MACH-03:** The system shall provide the option to schedule a time, only if the user is logged in.
    - **REQ-MACH-04:** Passwords shall be limited to a minimum of 8 characters and cannot be entirely numeric.

5. Terminology

- **Condominium:** a living complex in which the gym is located and the users reside.
- **Resident:** a person who lives in the condominium, the main user of this application.
- **Administration:** the condominium's administrative committee.

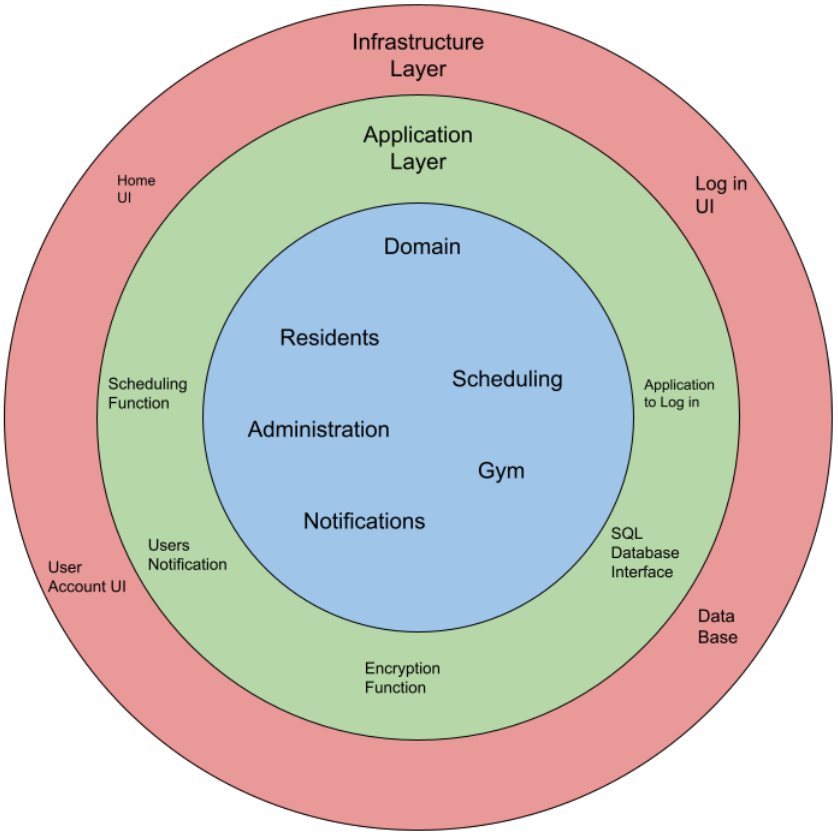
- **Domain Entities**
  - **Resident:** the person who will be scheduling in order to access the gym.
  - **Scheduling:** booking a time frame, in this case the gym.
  - **Gym:** the space that will be reserved when scheduling.

- **Domain Events**
  - **A scheduling has been made:** the user has booked a time frame in order to use the gym.
  - **A scheduling has been changed:** the user has changed the scheduling for a time different than before.
  - **Gym is full:** Happens when the maximum capacity of the gym has been reached, meaning that no more users can schedule for that time.

- **Domain Behaviors**
  - **Scheduling creation:** a user has made a scheduling, as such a space in the gym has been secured to them.
  - **Gym is full:** The gym capacity is at maximum therefore no more bookings can be made.

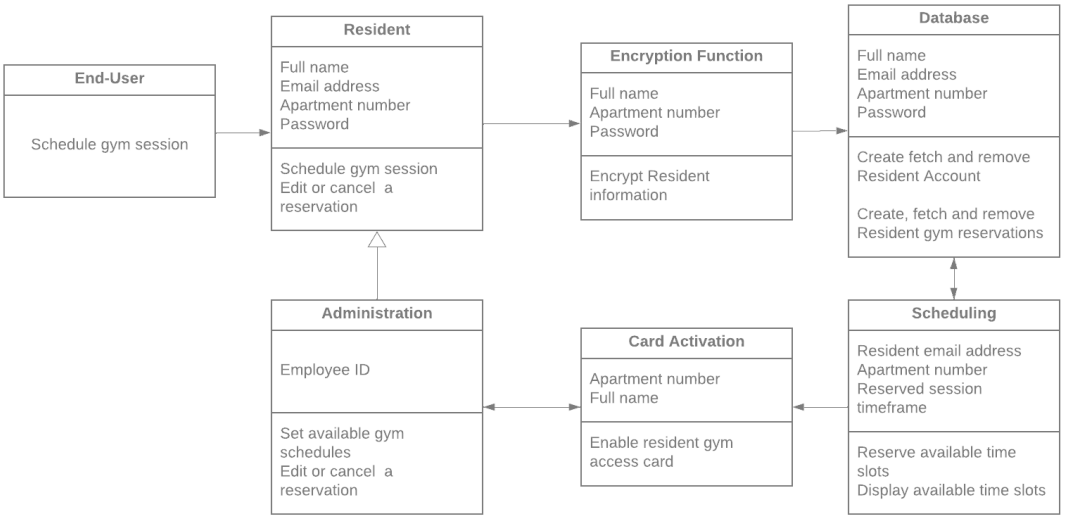
- **Domain Functions**
  - Once the maximum number of people have scheduled a certain time of the day, no more users can schedule for that time.
  - If a schedule is canceled, it will become available to other users.

6. Software Architecture Design



Using the Onion Architecture helps us separate all of the layers of the application. This helps exteriorize the database and other dependencies, by not building and implementing code around them. In doing so, it makes it easier to change the database and other dependencies, because the other layers are not dependent on the infrastructure layer. These layers build on each other and use different terminologies, the dependencies between them flow inward. For example, in this case the application layer will not be dependent on anything other than the domain layer. Evidently, the infrastructure layer will depend on both of the layers mentioned before. The domain layer contains the basis of the application, the entities are the following: Residents, Scheduling, Administration, Gym, and Notification. Following will be the application layer, this contains all of the business logic. This includes the scheduling function which will add and remove from a specific scheduled time. Finally we have the infrastructure layer, which contains the database and all of the UI components.

7. Software Component Design



Every component of the system is designed in a way where we can interact with the schedule. The interaction is what gives the purpose to the multiple functions of our application. While the End-User is an entity of the system, as they can schedule a session, edit a session. The database works as a storage for resident accounts and their attributes.

8. Selected Fragments of Implementation

```
export function getCurrentDate(separator=''){

  let newDate = new Date()
  let date = newDate.getDate();
  let month = newDate.getMonth() + 1;
  let year = newDate.getFullYear();

  return
`${year}${separator}${month<10?`0${month}`:`${month}`}${separator}${date
}`
  }
const currentDate = getCurrentDate();
const schedulerData = [
  { startDate: '2021-11-27T12:00', endDate: '2021-11-27T13:30', title:
'Go to gym' },
];
```

Function displays the current reservation for the user.

III. Analytical Part

- **Concept analysis of rough sketch**
  - The concept of “scheduling” in terms of the web application can be defined as the time reservation selected by the “user”. Every “resident” who wishes to use the “gym” must create an account. The user will be able to schedule a specific time frame of the day to use the gym facility, and cancel or edit their current “reservation”. The available timeframes will be displayed in chronological order.

- **Analysis of technology choices**
  - Django is the main choice for the implementation of this application. The simplicity of the database management makes Django a strong choice since saved efforts on database creations can be expended somewhere else. Since Django makes use of Python, Pytest is a safe choice in order to test the application, since it can run tests in parallel, execution times are reduced. As for the frontend, React is the top choice since it offers a lot of flexibility and customization.

- **Verification**
  - With the current development, the team decided to do individual tests on everyone’s system and see how the application works with different sizes of screens. This is because most of the development done was in the front end of the project. With the implementation of a front-end style, we intend to check if the dimensions of the web app are correct and that elements do not overlap with each other. While checking the front-end implementation we will test the account creation and the storage of its account information in the database. This will be done as well with each member’s system to test the functionality of the database as well and see live changes when accounts are created. After each member does the tests, the team will make a list of the implementations that need improvement and more functionality updates. Afterward we send our progress to the stakeholders to verify if the development is going according to their

specifications. After receiving their report, we repeat the process until we reach an agreement that the application is ready for deployment. This way the development made in each phase will be polished before moving into the final phase, where we will implement the final features.

- **Validation**
  - The team will maintain constant communication with the stakeholders. Maintaining code consistency will be a top priority to ensure that the system meets the requirements of the stakeholders. With residents testing we will ensure the interface is accessible. At the same time, with the application demos we will test along with the users to ensure that the application functions properly by making surveys that ask for specific feedback to be able to make the UI as frustration free as possible. Also we can ensure the quality of the application by making various beta tests to ensure that the functions work properly.

IV. Key Requirements

- **User Requirements**

1. The ability to change the amount of people allowed in the gym on the fly, in order to comply with unforeseen last minute restrictions on attendance capacity.

2. Allow users to make schedules in the hours that are available, to avoid scheduling conflicts.

3. Present users with information on the gym scheduling, so that users may be better equipped to use the system.

4. Allow administrators to freely change or cancel a scheduled gym appointment if the tenant does not show up, to make available that time slot for other users and optimize the facilities' use.

5. The ability to distinguish whether or not a person is allowed to enter and schedule an appointment, ensuring safety and better performance results for the system.

6. The system-to-be shall not allow users to schedule multiple time frames of the day.

7. The system-to-be shall allow the user to change or cancel the scheduled time.

8. The system-to-be shall allow the user to notify if they finished before the time scheduled, and send a notification to the next user in line.

9. The system-to-be shall order the available time frames in chronological order.

- **Software Requirements**

1. The system-to-be should update the slot availability in real time.

2. The system-to-be should not allow the reservation of the same slot by multiple users.

- **Business Requirements**

1. Allow administrators to ban use of the system-to-be when a person owes maintenance cost fees, in accordance with the current Horizontal Property Laws and Regulations of Puerto Rico.

2. The system must keep private information secure and inaccessible to third parties, in compliance with privacy laws and to avoid legal liability.