Table of Contents

1. Project Title and Authors

Our assigned team number is 28. The cool team name is: TEAM28. Our group members are Dongming Shen with USCID 7170243072, Tiancheng Xu with USCID 1522206865, Yi Jiang with USCID 5261852930.

2. Preface

The USCRecApp is an Android application designed for USC students to make reservations on USC recreational centers online. This document provides a requirements specification for this application.

3. Introduction

The USCRecApp allows USC students to make reservations before accessing recreational centers, under USC's guidance. Students should be able to view each USC recreational center on the map. Students should be able to book an available time slot or join a waiting list if the time slot is full. Students should be able to track their booking information and cancel an upcoming reservation. The USCRecApp would be more user friendly for mobile users than the current web reservation system, with extra msap and notification functionality.

4. Glossary

USC account: we assumed USCRecApp has access to each potential user (USC students)'s existing username, password, USCID, and a profile photo.

The system: when mentioning "the system," we are referring to the USCRecApp for simplicity

[button]: all buttons will be represented in this format

$Page$: all pages will be represented in this format

Direct (as a verb): when mentioning the system direct the user, we are referring to which page the user will end up to be, as determined by the system.

Response time: when a user triggers an event, like clicking a button, response time measures the time it takes for the event to finish.

Data corruption: data corruption is the error in data that occurs when user writes, deletes, and updates the database.

5. User Requirements Specification

User Story 1: Login
Description: Users should be able to login with their existing “USC account”.

User Story 2: View profile
Description: User should be able to view their profile information

User Story 3: View recreational centers on map
Description: Users should be able to view the location of all USC recreational centers on a map.

User Story 4: Check time slots availability
Description: Users should be able to check upcoming available time slots of each recreational center.

User Story 5: Book available time slots
Description: Users should be able to book available time slots for each recreational center.

User Story 6: Join waitlist and receive notification
Description: Users should be able to join the waitlist when a time slot is not available, and users in the waitlist should receive notifications when that time slot is available again.

User Story 7: View booking information
Description: Users should be able to track their booking information (view upcoming and previous reservations).

User Story 8: Cancel upcoming reservations
Description: Users should be able to cancel upcoming reservations.

6. System Requirements Specification

a. Functional Requirements

Title: Login - in $Login Page$

1. A user opens the application for the first time
2. The system should direct the user to Login Page
3. The user enter their username and password
4. The user press the [login] button to login
5. The system should check if the user enters a correct username and password
   5.1. The username/password are is incorrect
   5.2. The system should prompt the user an error message
   5.3. The system should keep the user in the login page
6. The user successfully logins
7. The system should direct the user to the Main Page (with the map)

Title: View profile - in $Profile Page$

1. The user clicks the [profile] button on the Main Page
2. The system should direct the user to the Profile Page
3. The Profile Page displays the student name, student USC id, and student photo
4. User can views the information in this page
5. The user clicks the [return] button to quits Profile Page
6. The system should direct the user to the Main Page (with the map)

Title: View recreational centers on map - in $Main Page$

1. The user enters the Main Page
2. The system should display all USC recreational centers on a map in the Main Page.
3. The user clicks a recreational center
4. The system should direct the user to the Booking Page of that recreational center

Title: Check time slots availability - in $Booking Page$

1. The user enters the Booking Page of a recreational center.
2. The system should display all time slots for that recreational center
3. Each slot should have an information box displaying its time and remaining available spots for the user to check
   3.1. The user clicks the [return] button to quits Booking Page
   3.2. The system should direct the user to the Main Page (with the map)
4. There should be a [make reservation] button next to the time slot.
   4.1. When a time slot has no available spot, the system should display a [remind me] button instead of a [make reservation] button next to that time slot
   4.2. The user click [remind me] to join a waiting list
   4.3. The system should add the user into the waiting list for that time slot
   4.4. The system should check and update availability of each spot periodically
   4.5. The system should notify the user by sending a notification to the pyhone board if the time slot becomes available again
5. The user click [make reservation] button to make a reservation
   5.1. When the user click the button, if no spot is available (someone else took the last spot before the user and the page did not refresh yet), the system should display an error message telling the user
   5.2. The system should then refresh the page and direct the user back to the same page
6. The system should add this reservation to the user's upcoming reservation list
7. The system should prompt a message telling the user the reservation is complete

Title: View booking information - in $Summary Page$

1. The user clicks the small window with reservation information on the Main Page
2. The system should direct the user to the Summary Page
3. The summary page displays the user's upcoming bookings and previous bookings
4. User can view the location and time slot corresponding to each booking on the page
5. There should be a [cancel] button for each upcoming booking to allow users cancel this booking
6. The user clicks the [return] button to quits Summary Page
7. The system should direct the user to the Main Page (with the map)

Title: Cancel upcoming reservations

1. The user click the [cancel] button next to an upcoming booking
2. The system should ask the user to confirm
   2.1. The user fails to confirm

    2.2. The system should not cancel the booking
3. The user confirms to cancel the booking
4. The system should cancel the corresponding reservation for the user
5. The system should remove the reservation information on the summary page
6. The system should notify other users in the waitlist of this timeslot (if any)
7. The system should prompt a message telling the user the cancelation is complete

b. Non-functional Requirements -- at least two (2) non-functional requirements

Title: speed
Description: button event response time is within 1 second.

Title: size
Description: the application should be within 500MB

Title: robustness
Description: the probability of data corruption is within 1%

7. System Model—Process Diagram

- Context Diagram (see Figure 5.1 of the textbook for an example)
- Process Diagram (see Figure 5.2 of the textbook for an example)
- Use Case Diagram (see Figure 5.3 of the textbook for an example)

Controller
Service - Agent
DAO

Agent Class -...> implement Common Services Interface (corresponds to buttons & page跳转)

Attributes:	User unique id
Methods:	Login (called automatically when first time open app)
	Check profile (get called when click [profile])
	View all time slots in a gym
	Make reservation
	Join waitlist
	Cancel reservation
	View reservations

Authentication Class: (对应agent login) - ALVIN

Attributes:	user unique id
Methods:	authenticate login information

Profile Class: (对应agent check profile) -xtc

Attributes:	User unique id
	Student name
	Student usc id
	Student password
	Student photo url
Methods:	Getters
	Display profile

Recreational Center Class: (对应agent view all timeslots in a gym) -xtc

Attributes:	recreational center id
	Name
	Location
	Time slot id list
Methods:	DisplayAllSlots() //显示所有time slots的info (from databse)

Timeslots Class: (对应agent Make & Cancel & View reservation) jy

Attributes:	Time slot id
	start & finish time
	Total slots 此timeslots总可以放下的人数
	Available slots left 此timeslots剩余可加入的人数
	Waitlist: 此time slots等待人员
Methods:	Display timeslpot Info() //显示此time slots的信息
	AddUser(userid) // 当某个user make reservation的时候call
	DeleteUser(userid) // 当某个user cancel reservation的时候call
	Check availability helper()

Waitlist Class: (对应agent Join Waitlist) jy

Attributes:	Time slot id

array of users id in the waiting list

Methods: addUser(user id)

notifyWL(): 通知WL中所有人员 // delete时，updata avail如果prev=0, now>0, call

Reservation Class: (对应view all booking information) jy

Attributes: User id

Methods: display reservation information(time slot id)

display all reservation information()

Center main key - name, location

Time slot main key - center id, start t, finish t, WL

Waitlist_info main key - time slot id, user id

HERE

Identify the major components in your system:

User Interface Layer: Main Map API, Login API, Profile API, Booking API, Summary API

Main Service Layer: Agent (Activity Switch Helper + Common Service)

Application Function Layer: User Login, View Profile, View All Timeslots, Make Reservation, Cancel Reservation, Join Waitlist, View All Reservations

Database Layer: Recreational Center Table, Timeslot Table, Reservation Table, Waitlist Table, User Table

Map all system components to requirements

User Story 1: Login - handled by Login API, Agent, User Login, Database

User Story 2: View profile - handled by Profile API, Agent, View Profile, Database

User Story 3: View recreational centers on map - handled by Main Map API

User Story 4: Check time slots availability - handled by Booking API, Agent, View All Timeslots, Database

User Story 5: Book available time slots - handled by Booking API, Agent, View All Timeslots, Make Reservation, Database

User Story 6: Join waitlist and receive notification - handled by Summary API, Agent, Join Waitlist, Cancel Reservation, Database

User Story 7: View booking information - handled by Summary API, Agent, View All Reservations, Database

User Story 8: Cancel upcoming reservations - handled by Summary API, Agent, Cancel Reservation, Database

Provide an architectural configuration – the block diagram – 贴图

Provide the rationale for the architectural configuration

（**a**）**Explain the choice of components**

Each component in the User Interface Layer corresponds to displaying an application page we will be using for this app and provide the user with APIs for interaction. In particular: …

Agent in the Main Service Layer handles the event in the User Interface Layer and calls the collections of procedures in the Application Function Layer to handle these events in detail.

Each component in the Application Function Layer corresponds to the interaction with the database in order to achieve the functionality required in the user stories. In particular: …

All components in the database layer correspond to the tables we will be using to store specific information for the application, which will be requested when the application function layer is handling events.

（**b**）**Explain the choice of architecture style**

We chose layered architecture, because web application is a typical example of layered architecture. The application can be divided into several layers, in which each layer acts as a client and a server to the

layer nearby, and components inside each layer do not interact with each other. For instance, the application function layer asks the database layer for information, and the main service layer calls the collections of procedures in the application function layer. Therefore, it makes sense to choose layered architecture.

Preface

The USCRecApp is an Android application designed for USC students to make reservations on USC recreational centers online. This document has two purposes: First, it provides a high-level architecture of the whole system to describe how different system components interact with each other. Second, it provides an implementable detailed design (with both a UML class diagram and some UML sequence diagrams) to be used by software engineers to implement the system based on this design.

Introduction

The components in User Interface Layer are Main Map API, Login API, Profile API, Booking API, Summary API.

The component in the Main Service Layer is Agent, which corresponds to the Activity Switching Helper and Common Service in the UML class diagram.

In Application Function Layer, there are functionalities like User Login, View Profile, View All Timeslots, Make Reservation, Cancel Reservation, Join Waitlist, View All Reservations, which corresponds to the procedures implemented in the Agent class, like make_reservation, cancel_reservation, etc.

Database Layer has tables like Recreational Center Table, Timeslot Table, Reservation Table, Waitlist Table, User Table. The tables correspond to the other classes in the UML class diagram. For example, the Recreational Center class corresponds to the Recreational Center Table, Timeslot class corresponds to Timeslot table,etc.

Main Map API:

Login API:

Profile API:

Booking API:

Summary API:

Agent (Activity Switch Helper + Common Service): Agent class that implements the Common Service and Activity Switching Helper interface.

User Login: user_login(user_name: String, password: String) procedure inside Agent class

View Profile: view_profile() procedure inside Agent class

View All Timeslots: view_all_timeslots(unique_center_id: String) procedure inside Agent class.

Make Reservation: make_reservation(unique_timeslot_id: String) procedure inside Agent class.

Cancel Reservation: cancel_reservation(unique_timeslot_id: String) procedure inside Agent class.

Join Waitlist: join_waitlist(unique_timeslot_id: String) procedure inside Agent class.

View All Reservations: view_all_reservations() procedure inside Agent class.

User Table: the profile and agent class which correspond to a user.

Recreational Center Table: the Recreational Center class.

Timeslot Table: the Timeslot class.

Reservation Table: the Reservation class.

Waitlist Table: the Waitlist class.