Mavs Tech Rental

(MTR)

Course Name: CSE 5324 Software Engineering I: Analysis, Design, and Testing

Professor: Dr. Michael F. Siok

Course Number: CSE 5324-002

Team Group 6, Name - Maverick Developers

Team Group 6 Members:

- Aryan Verma (1002079331)
- Anuhya Patibanda (1001969235)
- Akshay Raina (1001789877)
- Mudra Patel (1001860645)
- Shubham Arya (1001650536) (Captain)

Date: December 5th, 2022

Iteration: 3

Table of Contents

Project Description

Most people do not have the resources to use in terms of both hardware and software needs technologically as it is sometimes too expensive for them to buy it. Some users have temporary needs for specific technology, so they need to rent it for a short period of time because purchasing it is not practical. Other users may want to try a device before making an informed decision about the hardware they plan on buying. Mavs Tech Rental as a Service solves this problem by allowing users to rent technology items according to their needs.

Authentication: The application allows users to sign in with their emails and passwords to access the content. Users that are not registered will be able to create an account with their emails and password. This gives users safe access to the application and provides security. Once the user is signed in, a user should be able to sign out of their account at any point. Once signed out, the user would need to sign in again to access their information.

View items: Signed-in users can see a list of technology items that are available for them to rent out. These items are the items that are not currently rented out by any other user. Once a user rents out an item, the item is removed from the list until it is returned by the same user. The list of items on the screen will show essential information about the item like its name, picture, company, type of device, and other hardware specifications.

Filters and Search: Users will be able to filter items based on different parameters making it easier to find the actual item from the available list. Users can search for an item with its name.

Rent item: On clicking any item on the list of items, the user will be able to view all details about the device. Here, users can select the dates they want to rent it out by using a date and time picker. Once the user has selected the dates, they want to rent it out till the user can tap on the rent button to complete this action. Here we assume that this action rents the item to the user.

Notification: Users will be sent notifications before their item is due as a reminder to prevent them from being late. They could customize the reminder times according to their preference.

Returning items: Admins will be responsible for marking the returned item as available after resetting the device's configuration back to original and updating the view items list. A late return of the item will result in a fine for the user which is reflected on their profile.

Status: Users can check the status of the rented item in the profile section. This will allow the users to extend their deadlines if they want to until another date.

Statistics: Users will be able to see their current rented items, along with a history of items rented previously, and any pending balance or fines they have on the app.

Resources to be utilized: Database (Firebase) with items being offered to rent, customer details etc., Android Studio, Phone or Simulator, Wireless Internet.

Team Members

Aryan Verma: I've had some experience with Flutter and have worked directly with web development. I haven't tried creating apps with Android Studio, but as the project advances, I want to. I love UI/UX design, am a skilled coder, and have made websites and applications in the past. I am aware of the many layers and the overall structure that should be present while creating a good application. In addition to overseeing my final project, I oversaw the coding club at my university.

Anuhya Patibanda: I have worked with several project groups in my graduate courses and undergraduate courses; therefore, I am familiar with designing UML and working with databases. I have also done application-based projects connected to the cloud utilizing the Java language. I am familiar with Android Studio, but I haven't used it for real projects. I hope to pick up some new skills as I construct this application.

Akshay Raina: I have 5+ years of experience in rapid custom application development using Design Thinking, worked in Agile and SCRUM methodologies, architected apps, services, landscapes in Cloud Platforms as a Lead Engineer. Also, developed complex integrations with ML, IIoT services, and worked heavily on documentation, etc. for web / mobile apps for various Industries across the globe. I have built various hybrid Android apps earlier, but native Android app development is new for me, and I am super excited to learn it.

Mudra Vithalbhai Patel: I am familiar with Flutter applications and have previously made a travel application for easier public transit. I have also taken a short online introduction course for Android app development through which I made a small food ordering application. I am proficient in Java programming and have worked on UML designing for several of my undergraduate database courses. I have been a leader for multiple group projects and worked as an event coordinator for college symposiums.

Shubham Arya: I have been an iOS developer for the past 2.5 years over which I have designed and made several iOS apps, three of which are currently published on the App Store. While I do not have any prior experience working in Android Studio, my experience with iOS app development, in general, should come into use for our team to make a successful, functioning, well-designed application. I also have prior experience in making UMLs and working on team projects that will come in handy.

Requirements

Req ID	Req Statement	Line reference
R1	The system shall provide authenticated login to the Technology Rental app.	8, 9
R2	The system shall provide for new users to create an account to use the app.	9, 10
R3	The system shall provide for users to log out of the app.	10, 11
R4	The system shall provide authenticated users with a view of the technology rental options available for renting.	14 to 16
R5	The system shall provide essential information like name, image, and type of device on the screen with all rental items.	16 to 18
R6	The system shall provide authenticated users a search option to search for item by title.	21
R7	The system shall provide authenticated users an option to filter items by the type of device.	20, 21
R8	The system shall provide authenticated users with all necessary information about a technology item when the user selects the item to rent it.	23, 24
R9	The system shall allow authenticated users to select the dates for renting the technology item.	24 to 26
R10	The system shall provide authenticated users with a notification reminder to return the item before their due date.	28, 29
R11	The system shall provide the authenticated user a way to view the status of their items from the profile section in the app.	35
R12	The system shall allow authenticated users to extend the deadline of the technology item if they want to use it longer.	35, 36
R13	The system shall provide authenticated users with the history of items they have rented out in the past.	38, 39
R14	The system shall provide the admin an interface to update the status of a returned item.	31 to 33

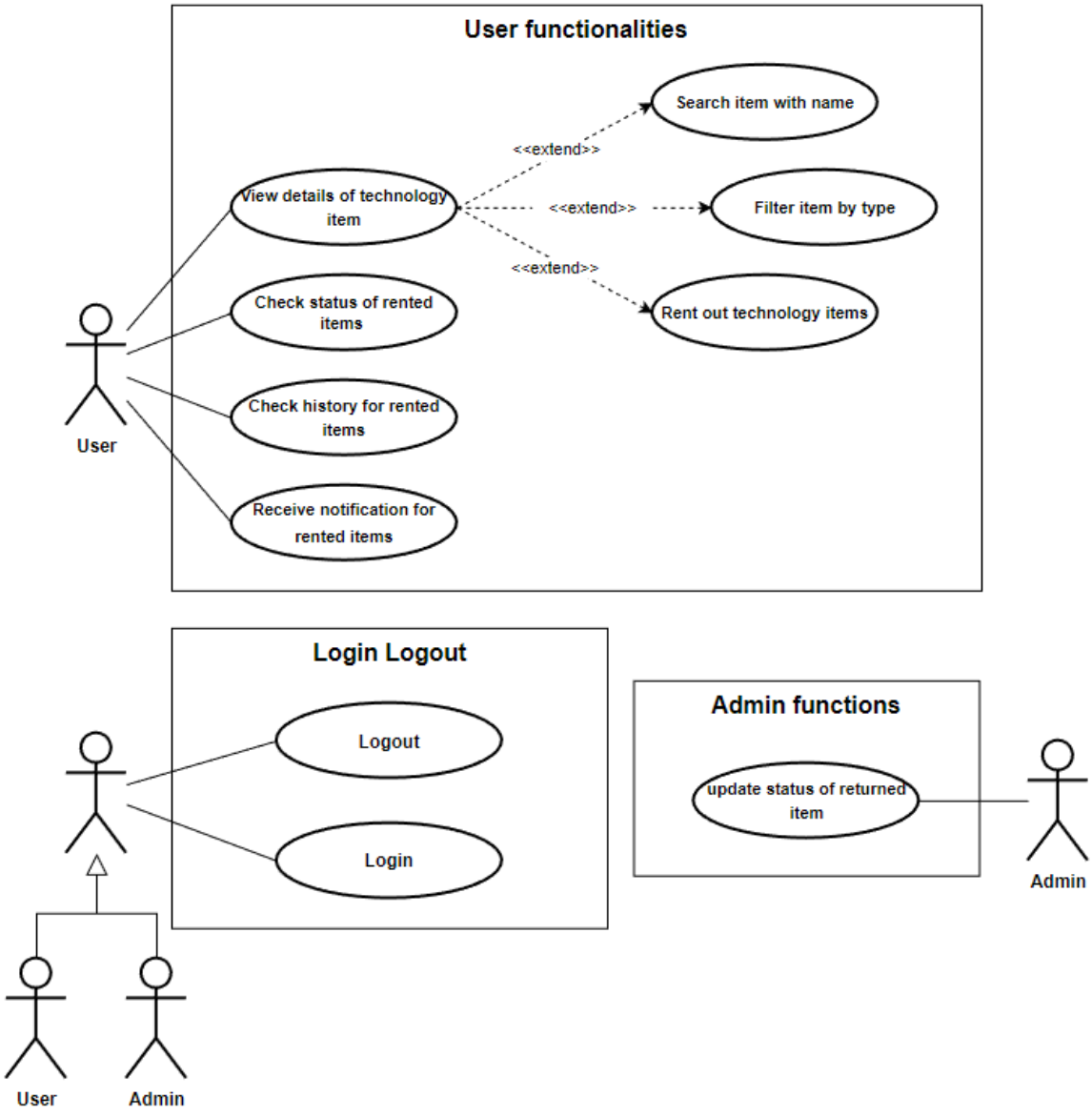
Use Case List

Use Case #	Use Case Name
UC 1	Login
UC 2	View Electronics
UC 2.1	Search Items
UC 2.2	Filter Items
UC 2.3	Rent Electronics
UC 3	Check Rental Status
UC 4	Check History for Rented Items
UC 5	Set Notifications for Rented Item
UC 6	Log out
UC 7	Update status of returned item

High Level Use Case

- **UC 1: Login**
  - TUCBW the user when they enter their login credentials.
  - TUCEW the user signed into the app and technology items home screen is displayed.
- **UC 2: View Electronics**
  - TUCBW the user clicking on a technology item from the list of technology items on the home screen
  - TUCEW the user viewing the full description and images of the technology item for them to see.
  - UC 2.1: Search Items
    - TUCBW the user clicks on the search button.
    - TUCEW the user seeing the list of the items that was searched.
  - UC 2.2: Filter Items
    - TUCBW the user clicking on the filter technology item button.
    - TUCEW the user seeing the filtered electronic items on the home screen.
  - UC 2.3: Rent Electronics
    - TUCBW the user selecting dates they want to rent it for.
    - TUCEW the user being rented the item they picked.
- **UC 3: Check Rental Status**
  - TUCBW the user clicking on the item they have rented out from their profile section.
  - TUCEW the user seeing the status of the item they clicked on.
- **UC 4: Check History for Rented Items**
  - TUCBW the user when they click on the check history button on their profile section.
  - TUCEW the user sees the list of their past rented items on the screen.
- **UC 5: Set Notification for Rented Item**
  - TUCBW the user rents an item.
  - TUCEW the user receives the notification on the day when the item is due.
- **UC 6: Log out**
  - TUCBW the user when they click on the log out button.
  - TUCEW the user being logged out and exited to the sign in screen.
- **UC 7: Update status of returned item**
  - TUCBW the admin searching for the user with the user's student ID.
  - TUCEW the admin clicking on the return device button.

Use Case Diagram



Requirements to Use Case Traceability Matrix

	Priority Weight	UC 1	UC 2	UC 2.1	UC 2.2	UC 2.3	UC 3	UC 4	UC 5	UC 6	UC 7
R1	1	X									
R2	1	X									
R3	1									X	
R4	1		X								
R5	1		X								
R6	3			X							
R7	3				X						
R8	2					X					
R9	2					X					
R10	4								X		
R11	5						X				
R12	10						X				
R13	5							X			
R14	2										X
	Score	2	2	3	3	4	15	5	4	1	2

Note: Priority 1 is the highest

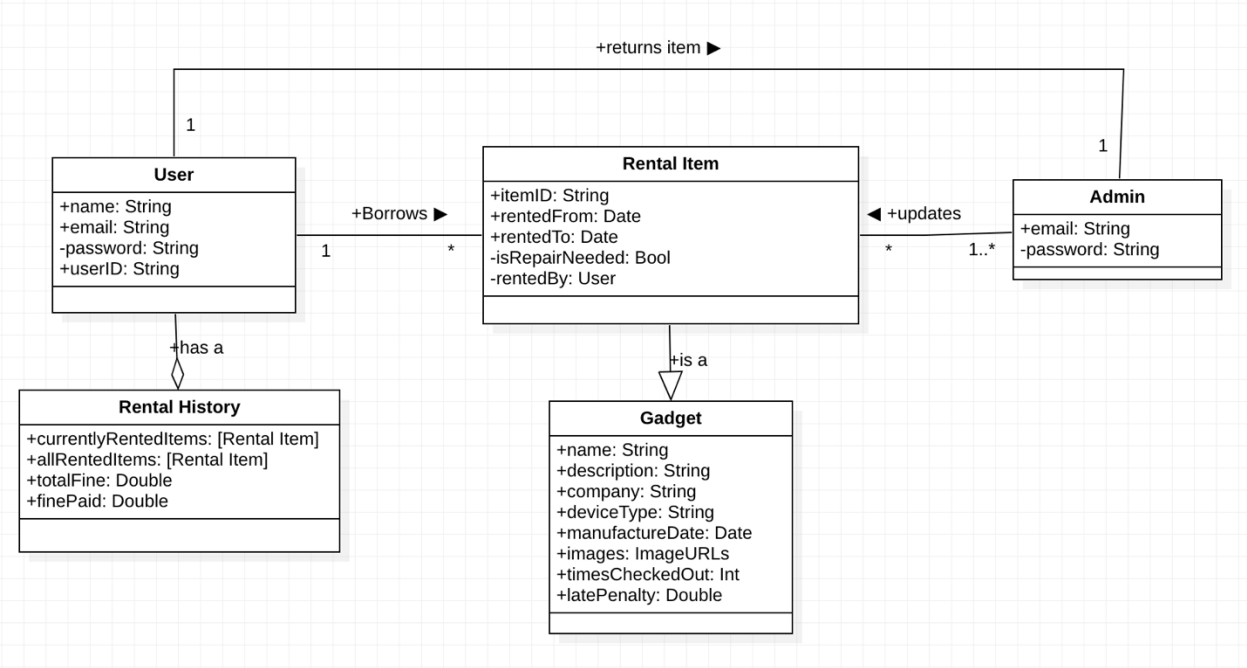
Increment Matrix

Use Case	Priority	Effort (person -weeks)	Depends on	Assigned to	Iteration 1 (10/07/22)	Iteration 2 (11/04/22)	Iteration 3 (12/05/22)
UC 1	**2**	3	None	MP, AP	3		
UC 2	**2**	3	UC 1	SA	3		
UC 2.1	**3**	2	UC 1, UC 2	SA, AV	2		
UC 2.2	**3**	2	UC 1, UC 2	SA, AV		2	
UC 2.3	**4**	4	UC 1, UC 2	AP, SA		4	
UC 3	**15**	3	UC 1, UC 2.3	MP, AV			3
UC 4	**5**	3	UC 1, UC 2.3	AR, MP			3
UC 5	**4**	1	UC 1, UC 2.3	SA, AV		1	
UC 6	**1**	1	UC 1	AP, AR	1		
UC 7	**2**	3	UC 2.3	MP, AR			3
Total Effort		**25**			**9**	**7**	**9**

1 person-Week = 5 hrs.

Team Members: Aryan V. (AV), Anuhya P. (AP), Akshay R. (AR), Mudra P. (MP), Shubham A. (SA)

Domain Model



Expanded Use Case

EUC 1: Login

UC1: Login	
Precondition: This use case assumes that the user/admin has not signed in with their email and password.	
Actor: User/Admin	System: Mavs Tech Rental app
	(0) System displays the login page
(1) TUCBW the user when they enter their login credentials.	(2) System processes the typed information check for correct format and length.
(3) The user taps on the login button when they are done typing their information.	(4) System processes the login credentials to check if a user with that email and password exists in the backend. (See Figure 1.1) (*)
(5) TUCEW the user signed into the app and technology items home screen is displayed.	
Post-condition: The user/admin has been logged in and can use all other features in the app.	

EUC 2: View Electronics

UC2: View Electronics	
Precondition: This UC assumes that a signed in user is on the home screen and wants to view more information about an electronic item.	
Actor: User	System: Mav Tech Rental app
	(0) System shows the list of all the available tech items on the screen.
(1) TUCBW the user clicking on a technology item from the list of technology items on the home screen	(2) System takes the user to another screen that displays complete details about that electronic item.
(3) TUCEW the user viewing the full description and images of the technology item for them to see.	
Postcondition: The user has all information they need to determine whether they want to rent that item	

EUC 2.1: Search Items

UC2.1: Search Items	
Precondition: This UC assumes that a signed in user is on the home screen and cannot find the electronic item from the list.	
Actor: User	System: Mav Tech Rental app
	(0) System shows the list of all the available tech items on the screen. (See Figure 2)
(1) TUCBW the user clicks on the search button.	(2) System displays the search bar that user can type on.
(3) The user stats typing the name of the product they are searching.	(4) System matches the written characters to the titles in the list and updates the home screen. (See Figure 2.1) (*)
(5) The user sees the changes as they type. When user is done typing, user clicks on the search button.	(6) System processes the final typed characters and finds matches from the list of electronic items in database. (See Figure 2.1) (*)
(7) TUCEW the user seeing the list of the items that was searched.	
Postcondition: The user has found the item they want to see.	

EUC 2.2: Filter Items

UC2.2: Filter Items	
Precondition: The signed in user must be on the home screen and wants to filter items according to their needs	
Actor: User	System: Mav Tech Rental app
	(0) System shows the list of all the available tech items on the screen.
(1) TUCBW the user clicking on the filter technology item button.	(2) System displays the filter menu with all the various options to filter items. (See Figure 2.2)
(3) User selects the filter option for a particular type of item.	(4) System matches the user's filter options and shows filtered items. (See Figure 2.2.1) (*)

(5) TUCEW the user seeing the filtered electronic items on the home screen.	
Postcondition: The user has found the filtered item they want to see.	

EUC 2.3: Rent Electronics

UC2.3: Rent Electronics	
Precondition: The signed in user has found their desired item, is on the item detail screen, and now wants to rent it.	
Actor: User	System: Mav Tech Rental app
	(0) System displays information about desired item.
(1) TUCBW the user selects dates they want to rent it for.	(2) System registers the rental period selected by user to check its validity.(*)
(3) The user clicks on the “Rent Device” button.	(4) System rents device to user, makes it unavailable for other users and goes back to home page. (See Figure 2.3 and 2.3.1) (*)
(5) TUCEW the user being rented the item they picked.	
Postcondition: The user has rented the desired item.	

EUC 3: Check Rental Status

UC 3: Check Rental Status	
Precondition: The signed in user is on the profile screen and wants to check the rental status of the item they rented.	
Actor: User	System:
	(0) System displays the profile of the user
(1) TUCBW the user clicking on the item they have rented out from their profile section.	(2) System displays the rental status of the rented item and the due date. (See Figure 3 and 3.1) (*)
(3) TUCEW the user seeing the status of the item they clicked on.	
Postcondition: The user has checked the rental status of their rented item	

EUC 4: Check History for Rented Items

UC 4: Check History for Rented Items	
Precondition: The signed in user is on the profile screen and wants to check his rental history for rented items	
Actor: User	System: Mav Tech Rental app
	(0) System displays the profile of the user. (See figure 4)
(1) TUCBW the user when they click on the check history button on their profile section.	(2) System displays the list of the previously rented items. (See figure 4.1) (*)
(3) TUCEW the user sees the list of their past rented items on the screen.	
Postcondition: the user has checked the history of the rented items	

EUC 5: Set Notification for Rented Item

UC5: Set Notification for Rented Item	
Precondition: The signed in user has rented item and is on the rented item screen and needs a reminder for when the item is due.	
Actor: User	System: Mav tech rental app
	(0) System displays rent page.
(1) TUCBW the user rents an item.	(2) System schedules a notification on the day the rented item is due. (See Figure 5) (*)
(3) The user clicks on the dismiss button on alert after reading it.	(4) System triggers a notification to the user on the rented item due date. (See Figure 5.1)
(3) TUCEW the user receives the notification on the day when the item is due.	
Postcondition: The user has received the notification of the due date.	

EUC 6: Log out

UC: Log out	
Precondition: The user must be signed in and must be on the profile screen.	
Actor: User/Admin	System: Mav Tech Rental app
	(0) System displays the profile page that also has the log out button.

(1) TUCBW the user when they click on the log out button.	(2) System processes the current user information to initiate a sign out.
(3) The user sees a loading indicator that tells them sign out is in process.	(4) System signs out user and displays the sign in screen. (See figure 6 and 6.1) (*)
(5) TUCEW the user being logged out and exited to the sign in screen.	
Postcondition: The user has successfully logged out of their account.	

EUC 7: Update Status of Returned Item

UC: Update Status of Returned Item	
Precondition: The admin receives the physical device from the user to return it.	
Actor: Admin	System: Mav Tech Rental app
	(0) System shows list of all users to the admin.
(1) TUCBW the admin searching for the user with the user's student ID.	(2) System shows admin the user that was search for. (See Figure 7)(*)
(3) The admin clicks on the user that is displayed on the screen.	(4) System displays user's current rentals and profile information.
(5) The admin clicks on the rental item that the user returned.	(6) System displays details of the device.
(7) TUCEW the admin clicking on the return device button.	
Postcondition: The admin has updated the status of returned item on Firebase.	

User Interface prototypes

Figure 1

Figure 1.1

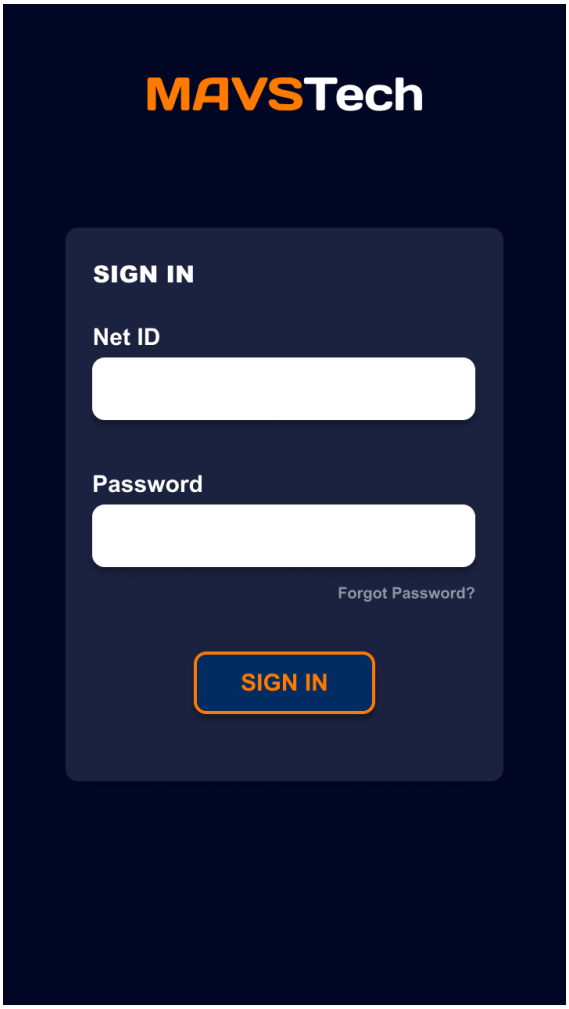


Figure 2

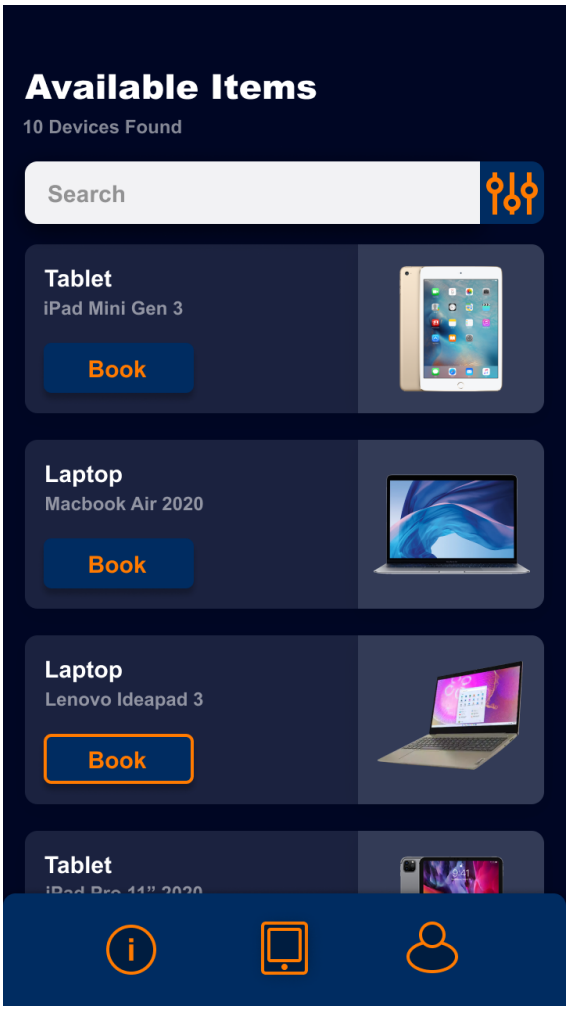


Figure 2.1

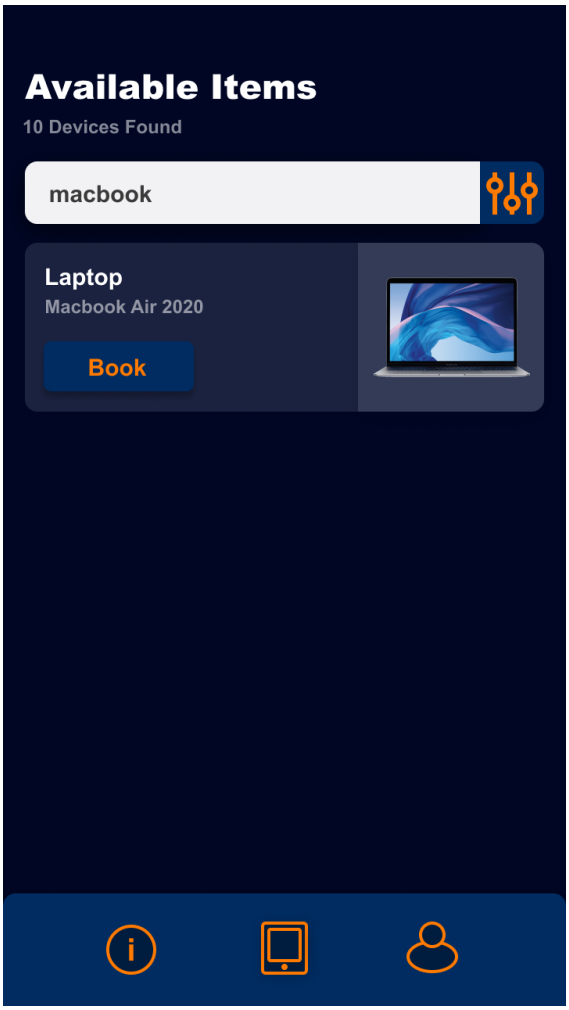


Figure 2.2

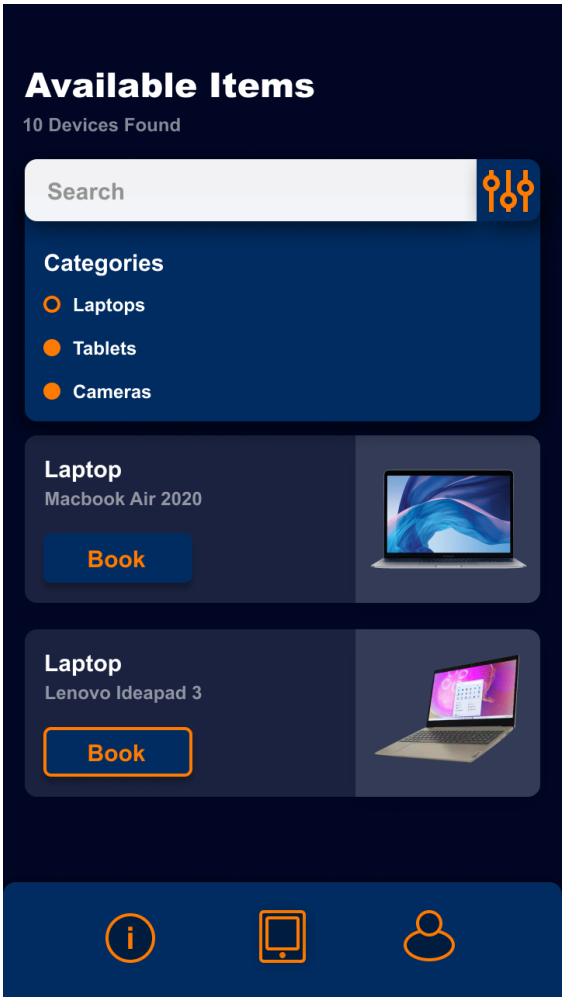


Figure 2.2.1

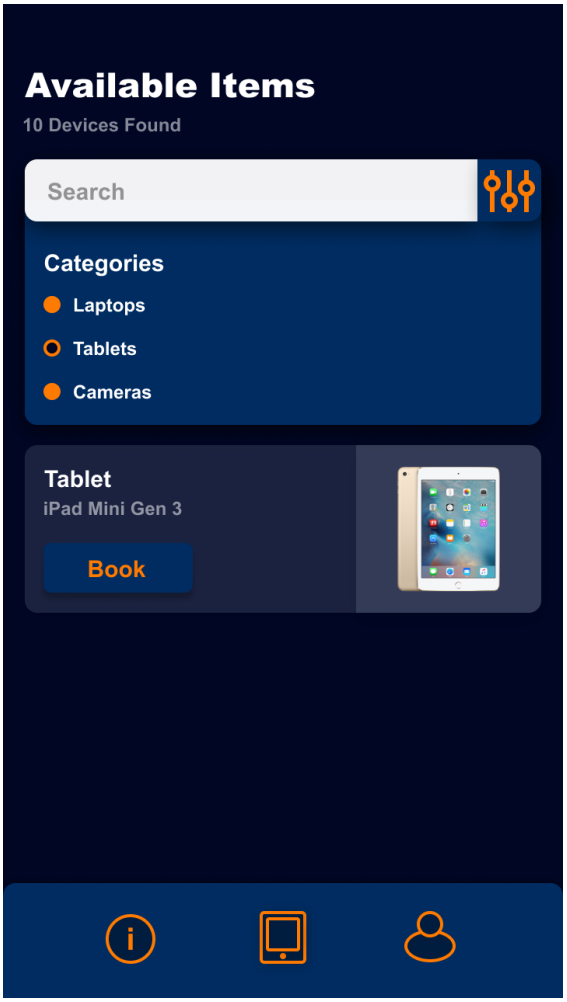


Figure 2.3

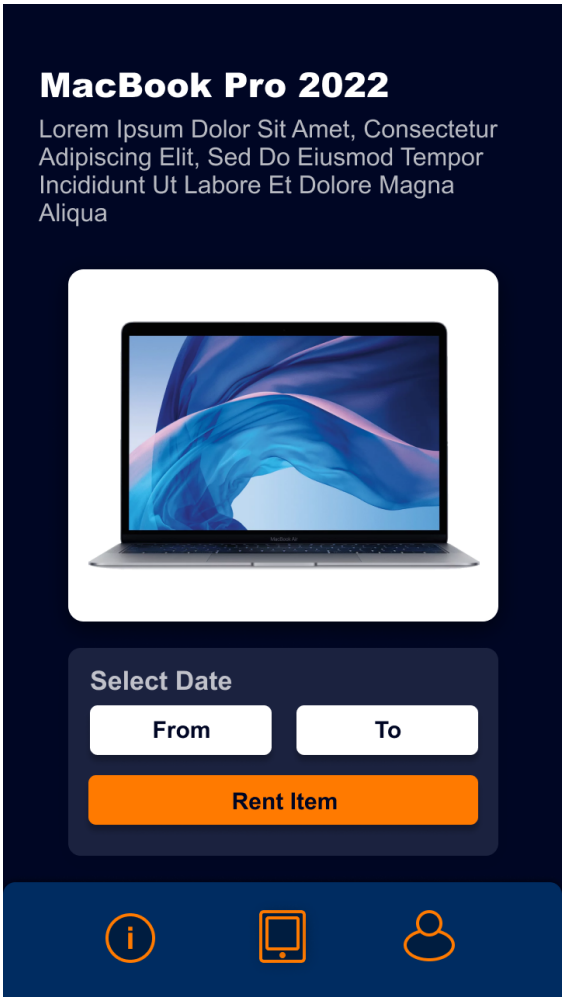


Figure 2.3.1

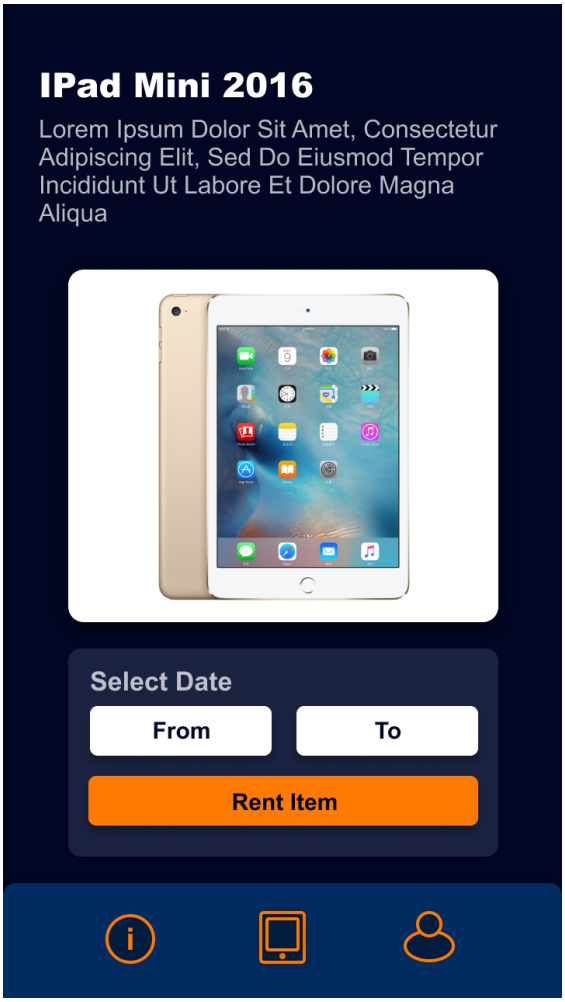


Figure 3

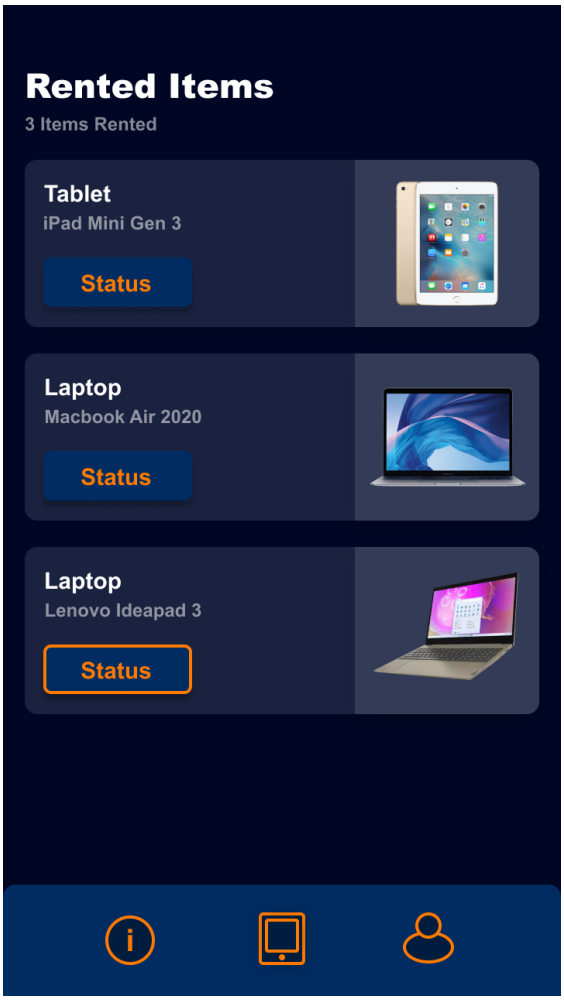


Figure 3.1

Figure 4

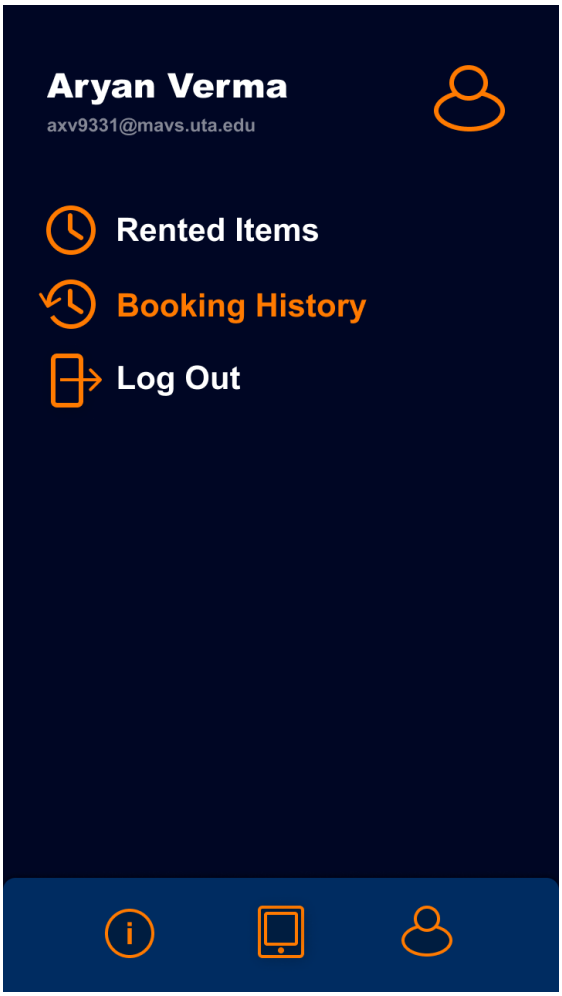


Figure 4.1

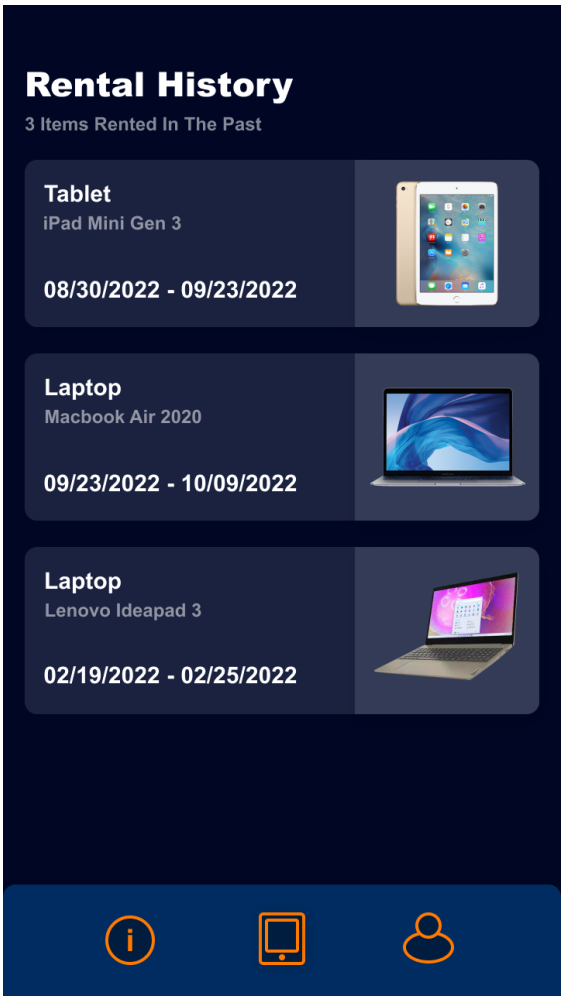


Figure 5

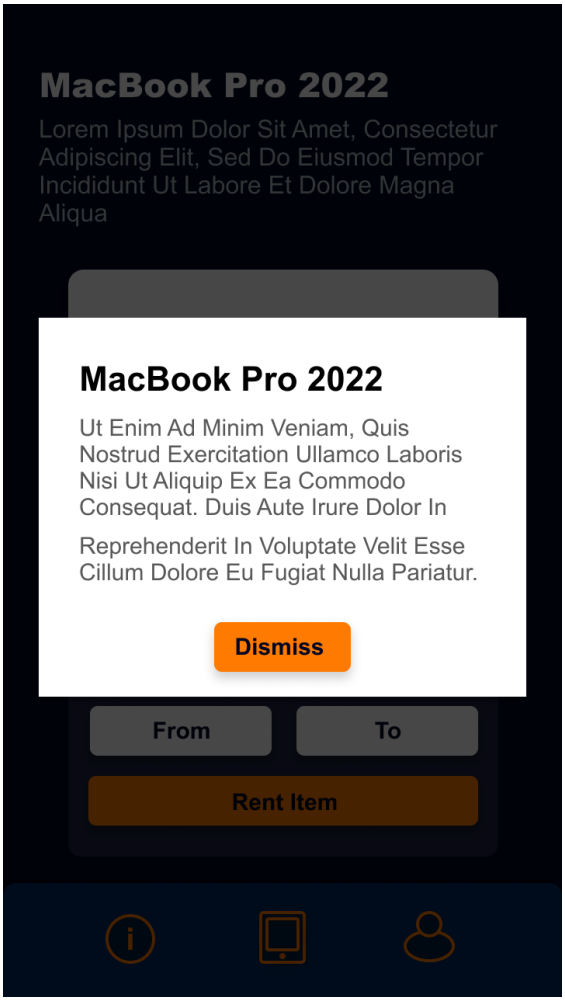


Figure 5.1

Figure 6

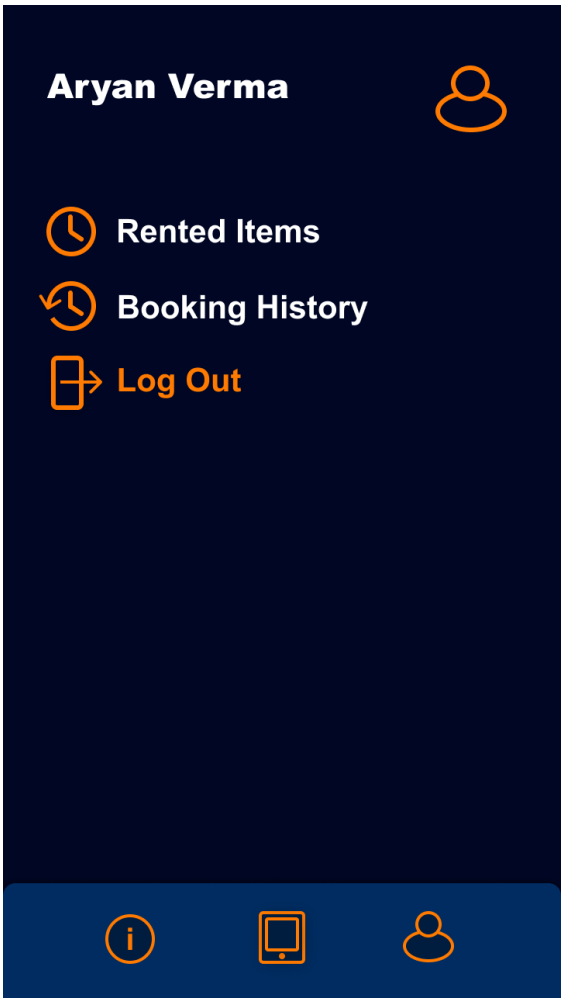


Figure 6.1

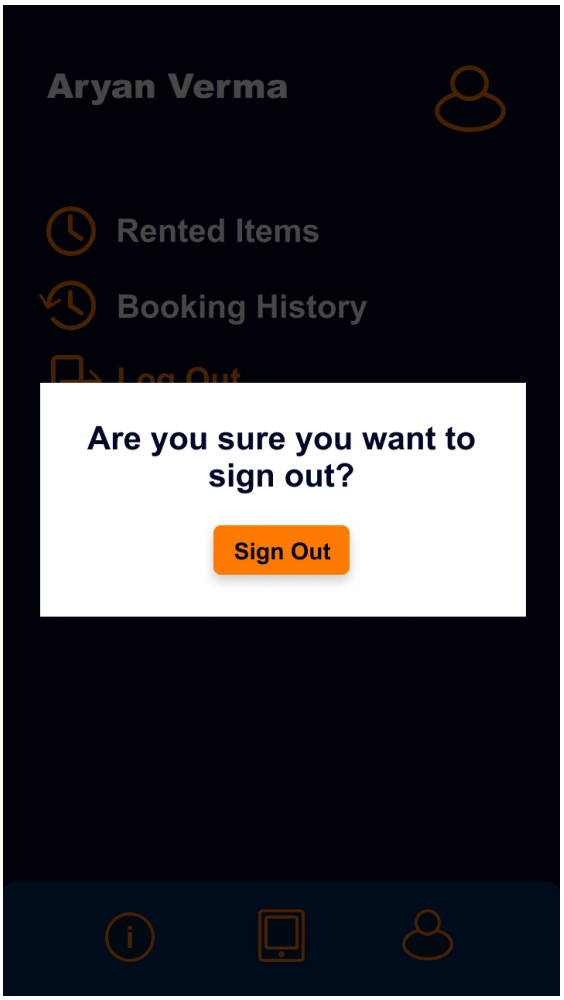


Figure 7

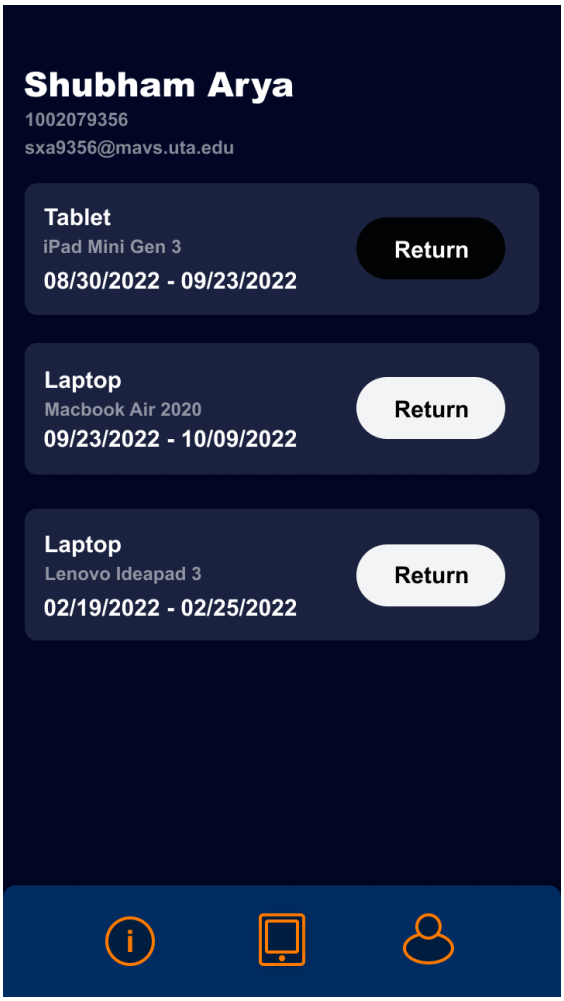
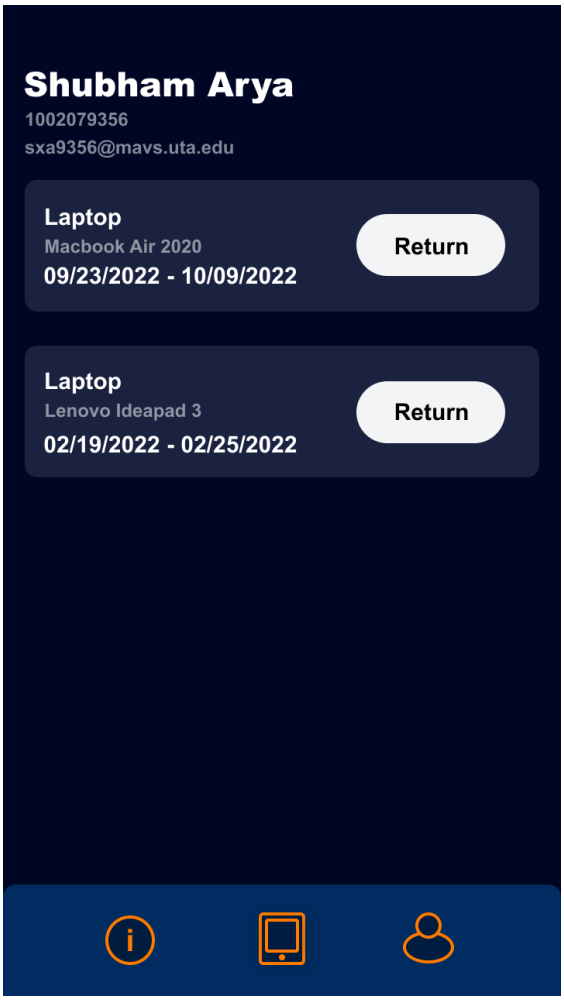
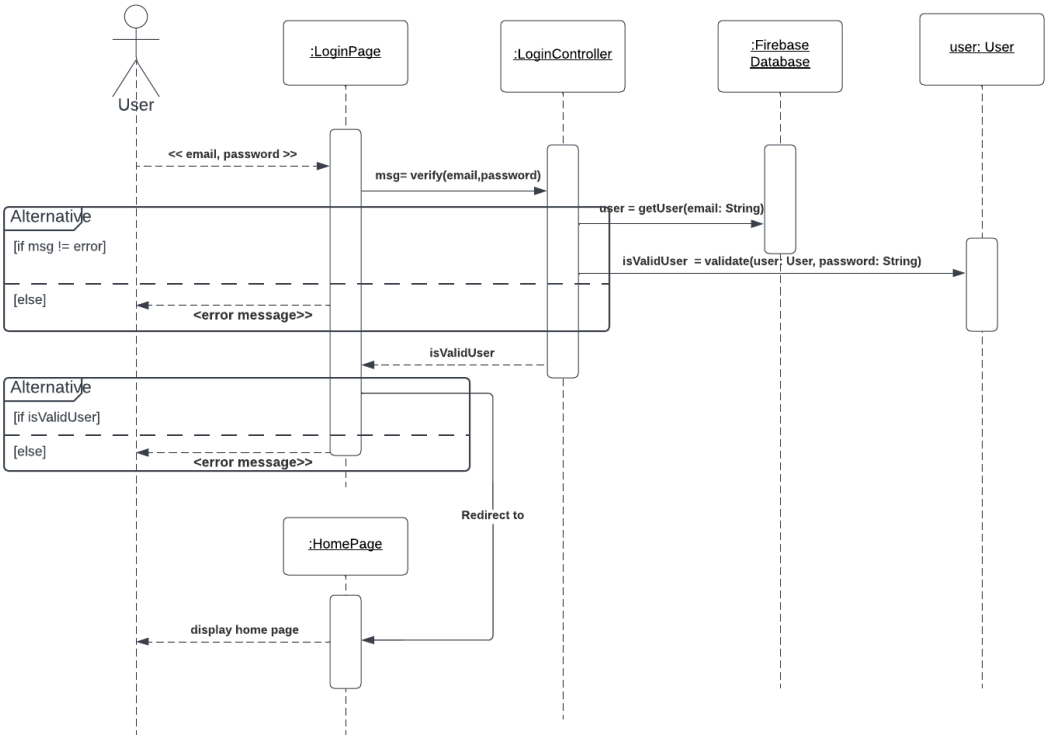


Figure 7.1

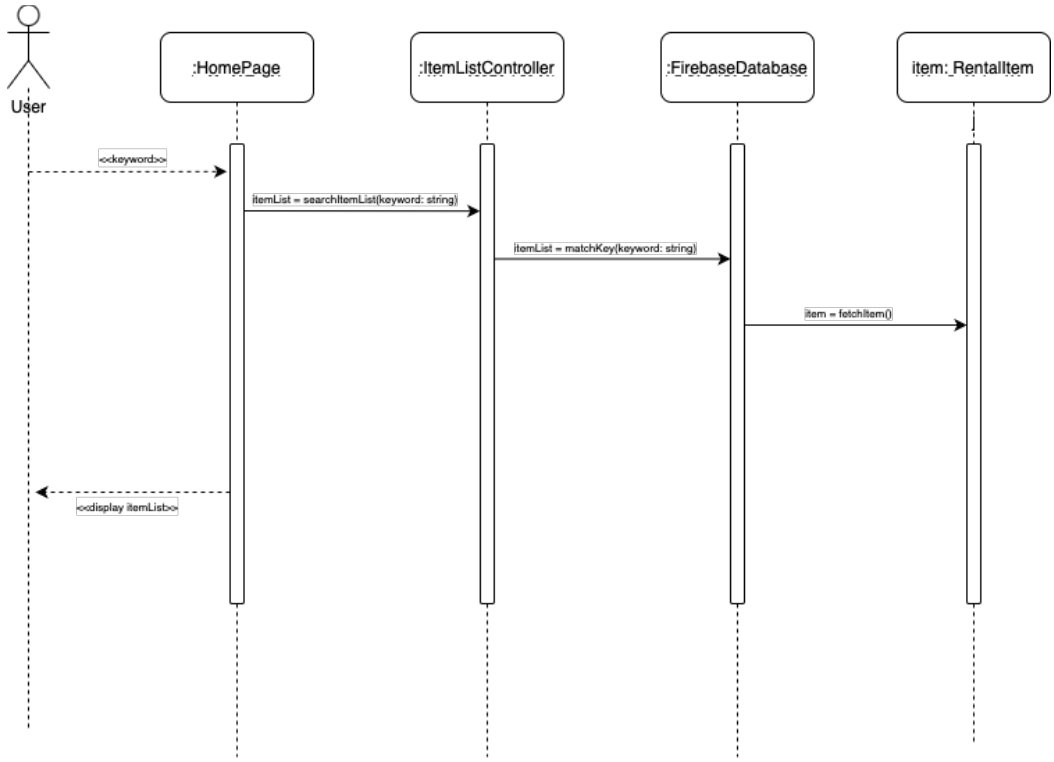


Design Sequence Diagrams

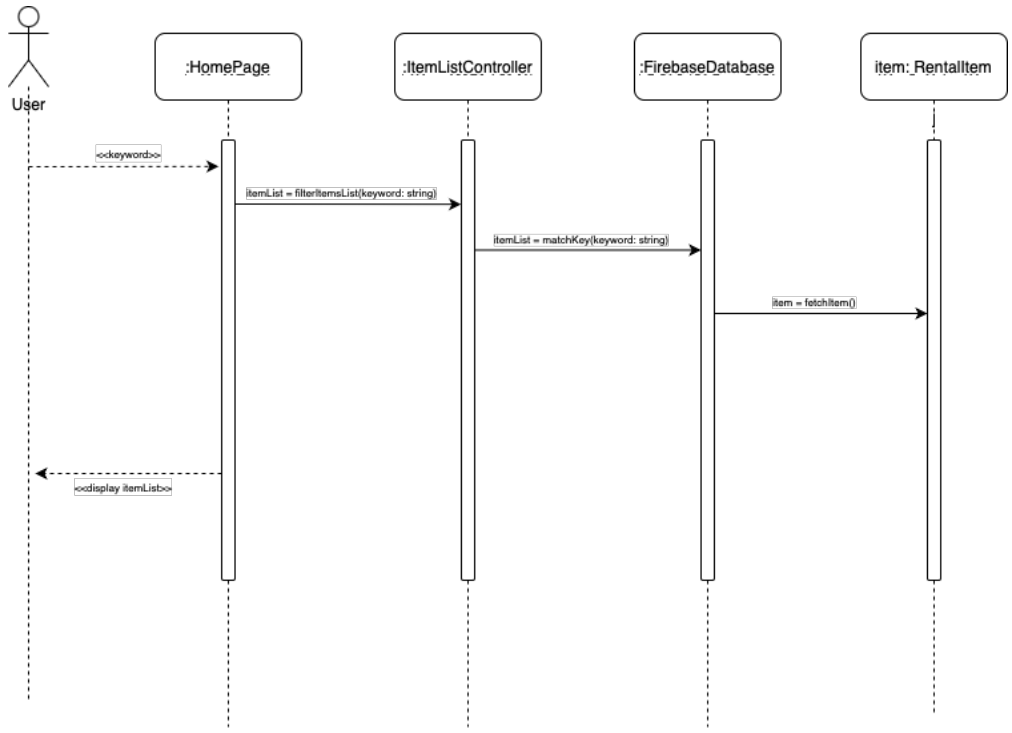
DSD 1: Login



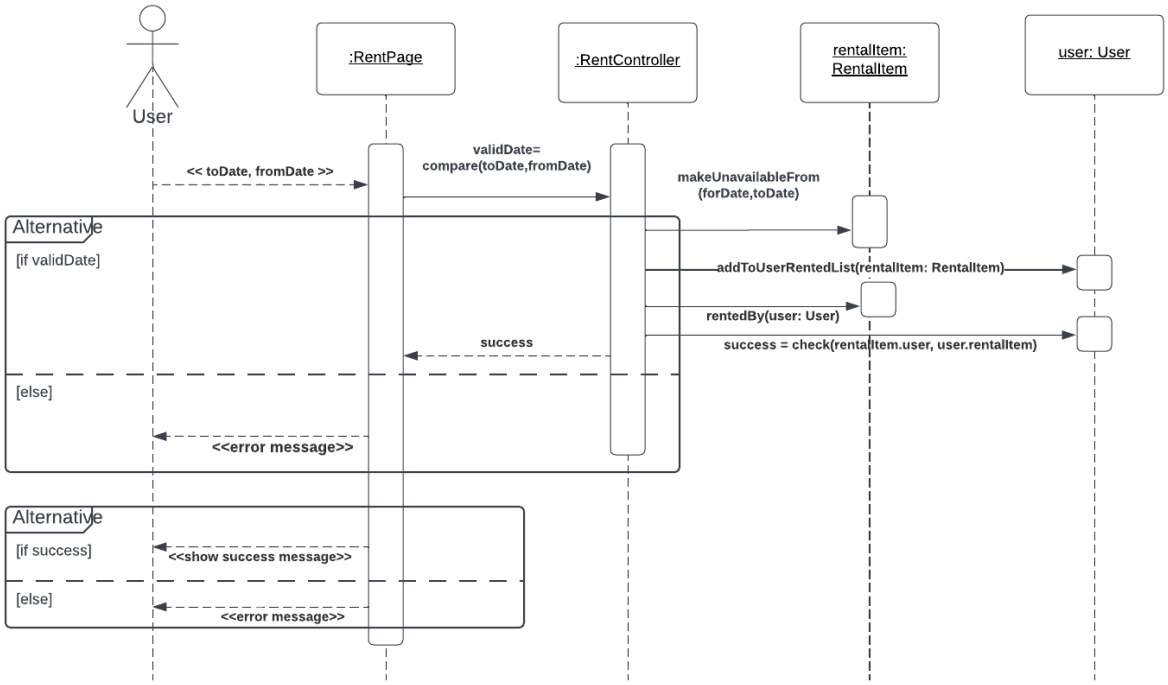
DSD 2.1: Search



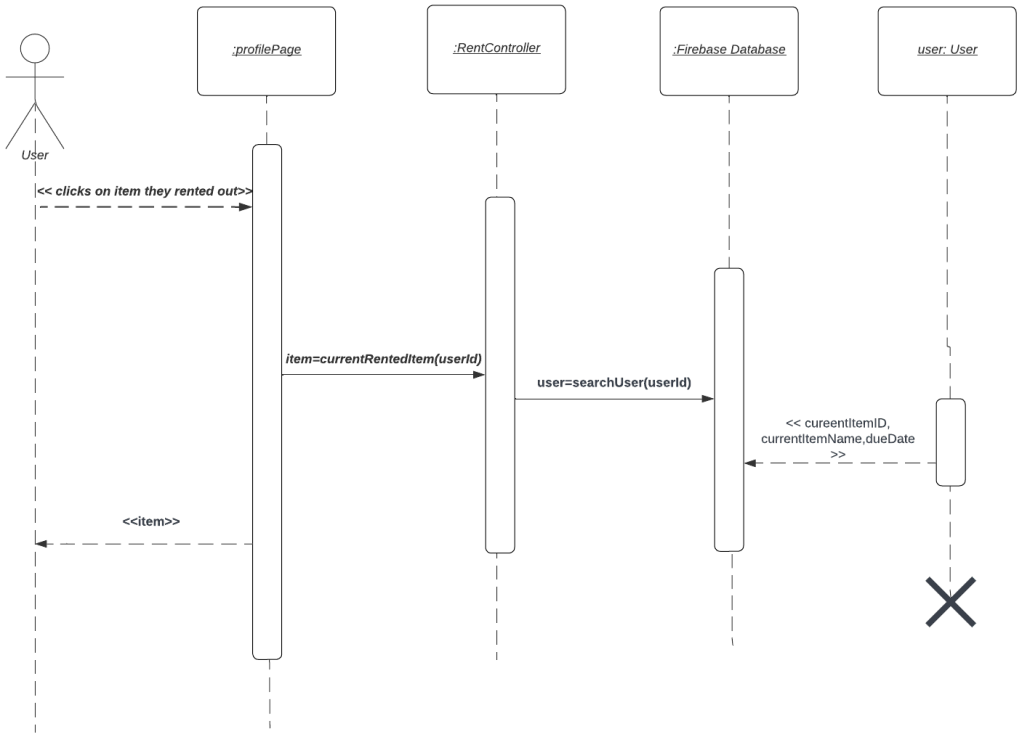
DSD 2.2: Filter Items



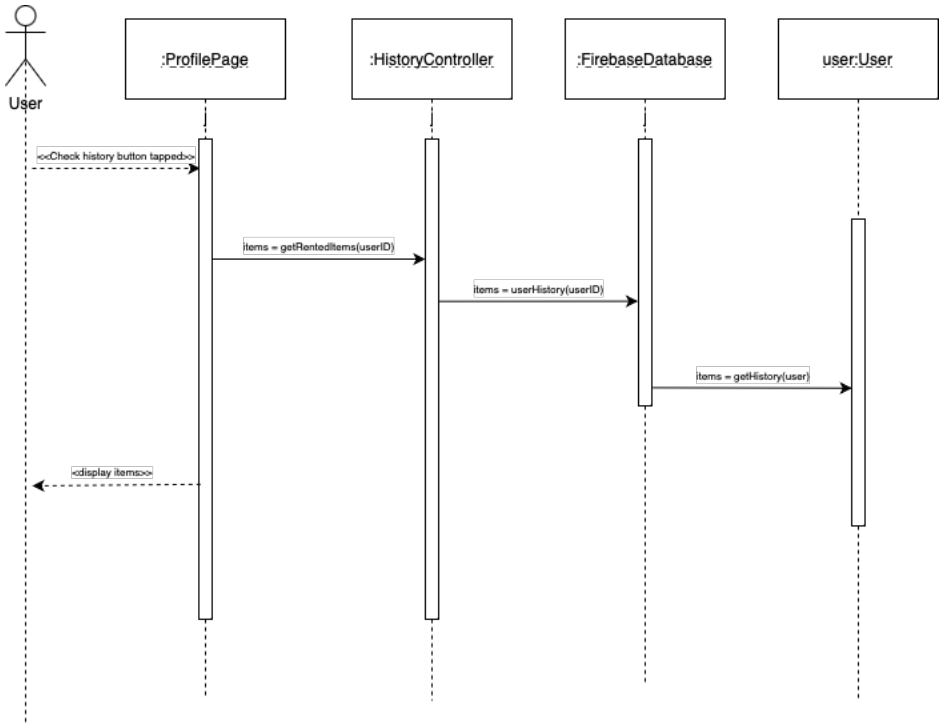
DSD 2.3: Rent Electronics



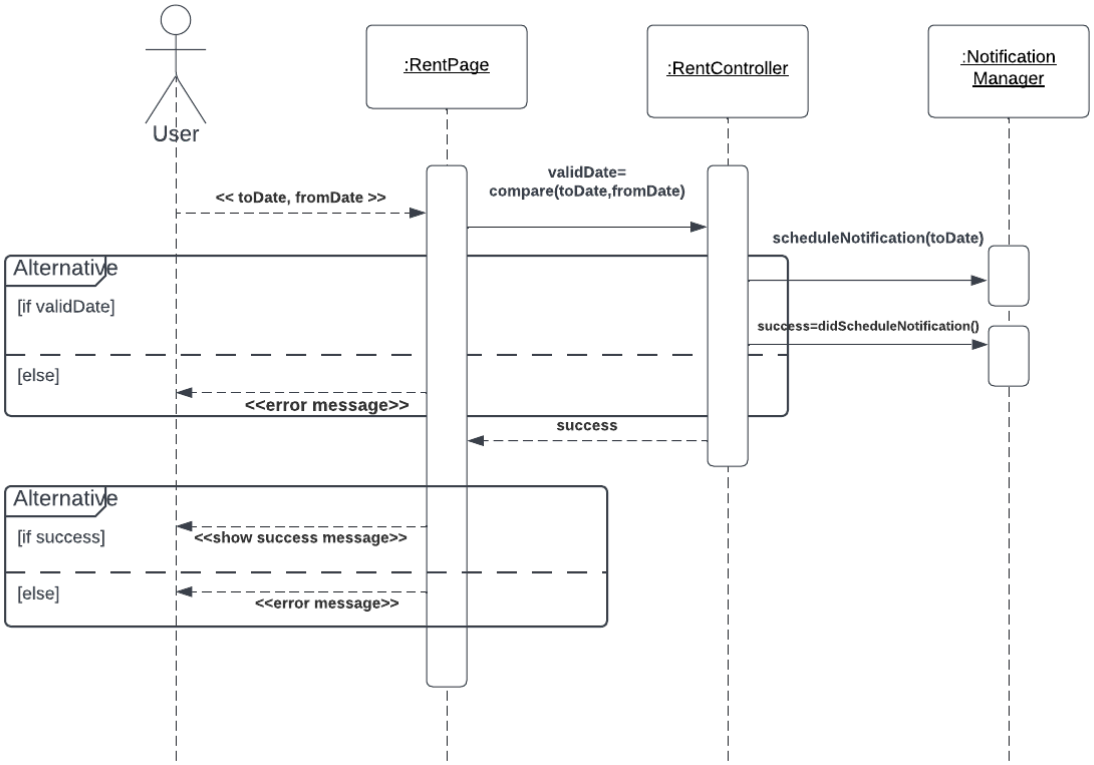
DSD 3: Check Rental Status



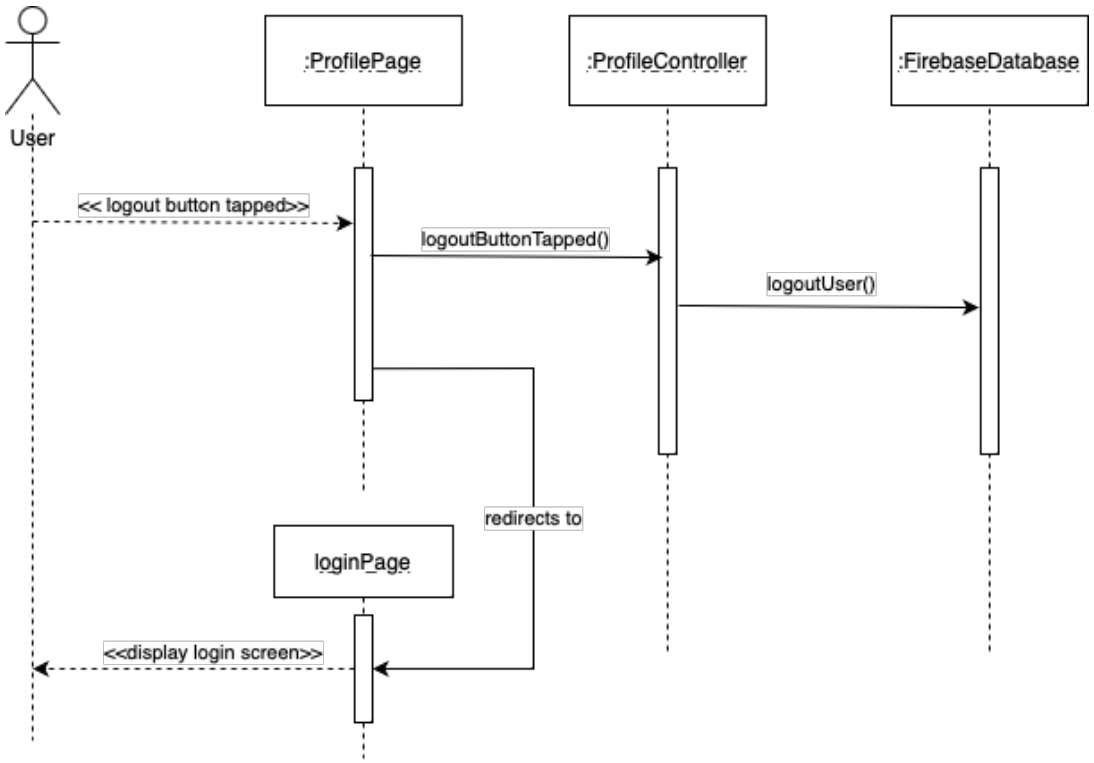
DSD 4: Check history for rented items



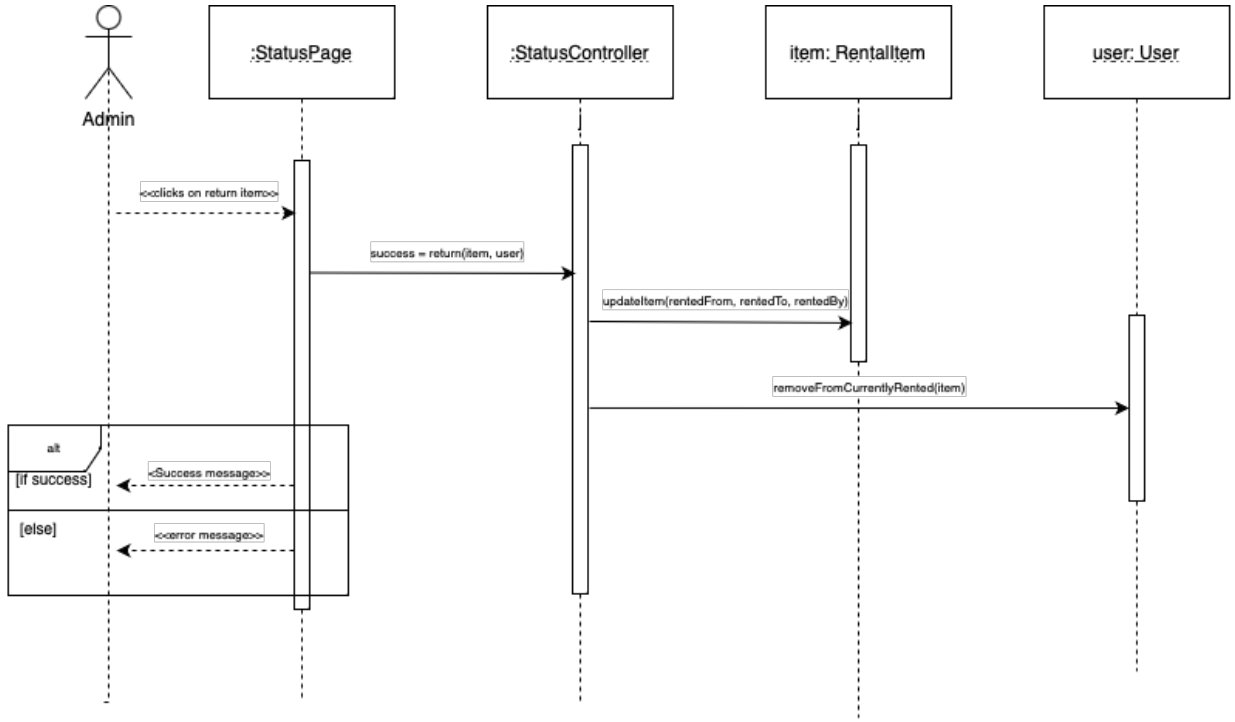
DSD 5: Set Notification for Rented Item



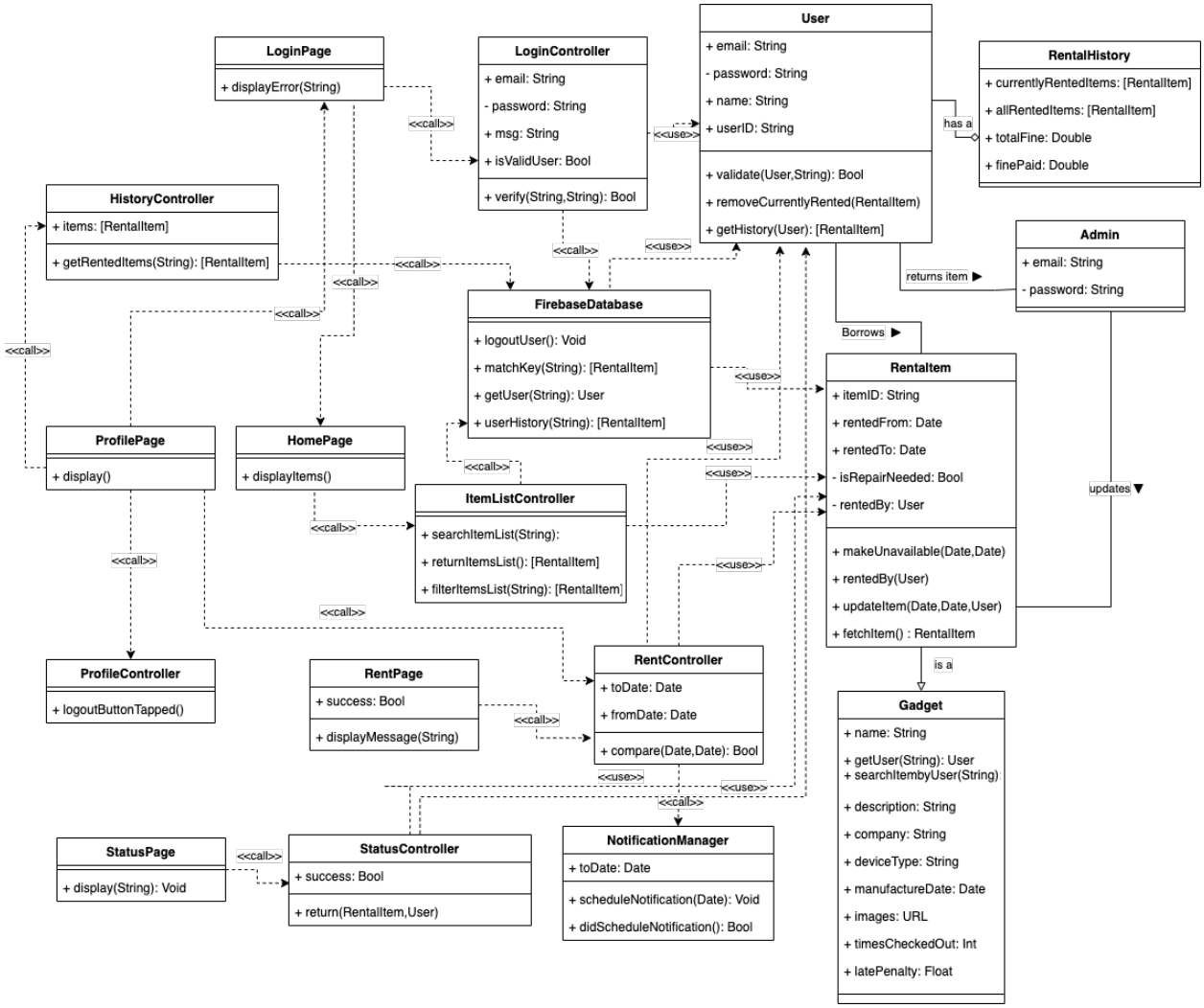
DSD 6: Log out



DSD 7: Update Status of Returned Item

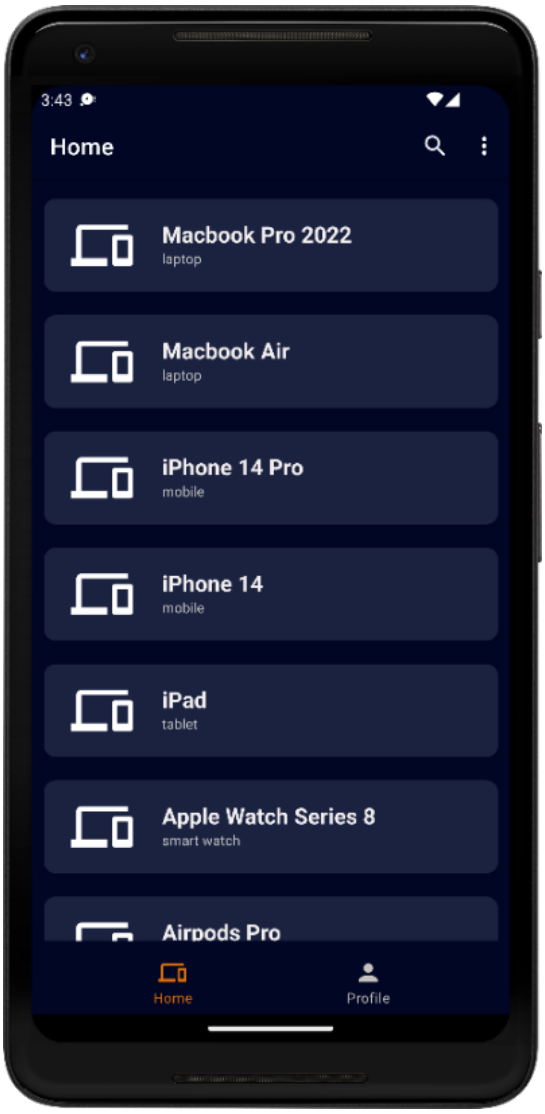


Design Class Diagram



Android Screenshots and Code

Displays all the items



Displays only search items

Code snippet:

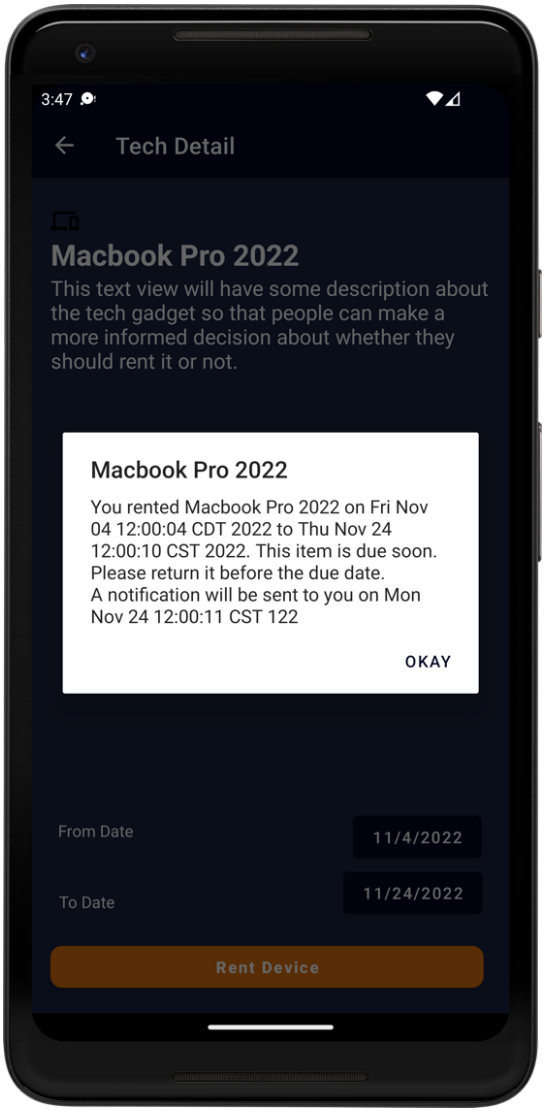
```
override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {

    inflater.inflate(R.menu.nav_menu, menu)
    val search = menu.findItem(R.id.nav_search)
    val searchView = search?.actionView as SearchView
    searchView.queryHint = "Search by name"

    searchView.setOnQueryTextListener(object : SearchView.OnQueryTextListener {
        override fun onQueryTextSubmit(query: String?): Boolean {
            return false
        }

        override fun onQueryTextChange(newText: String?): Boolean {
            val filteredDevices = devices.filter { device ->
                device.name.lowercase().contains(newText.toString().lowercase())
            }
            recyclerView.adapter =
                HomeRecyclerViewAdapter(filteredDevices) { selectedDevice: TechDevice ->
                    listItemClicked(selectedDevice)
                }
            return true
        }
    })
}
```

Schedule Notification



Screenshot of notification scheduled for date and time.

Code snippet:

```
private fun scheduleNotification() {
    val intent = Intent(applicationContext, Notification::class.java)
    val title = device?.name
    val message = "You rented $title on ${fromDate.toString()} to
${toDate.toString()}. This item is due soon. " +
            "Please return it before the due date."
    intent.putExtra(titleExtra, title)
    intent.putExtra(messageExtra, message)
    val pendingIntent = PendingIntent.getBroadcast(
        applicationContext,
        notificationID,
        intent,
        PendingIntent.FLAG_IMMUTABLE or PendingIntent.FLAG_UPDATE_CURRENT
    )
    val alarmManager = getSystemService(Context.ALARM_SERVICE) as
AlarmManager

    val notificationTime = Calendar.getInstance()
    notificationTime.set(toDate.getYear(), toDate.getMonth(), toDate.date,
12, 0)

    var time = notificationTime.timeInMillis
    alarmManager.set(
        AlarmManager.RTC_WAKEUP,
        time,
        pendingIntent
    )
    if (title != null) {
        alert(title, message+"\nA notification will be sent to you on
${Date(time)}")
    }
}
```

Application Demo

Here is the link for the Mav Tech Rental Demo on YouTube: https://youtu.be/JuhX0rw3d0U