Qatar University

College of Engineering

Department of Computer Science and Engineering

Senior Project Report

LabSpace

Project Group Members:

Fatima AL Suwaidi - 201800403

Fatma Jamal - 201608955

Fatima AL Kuwari - 201800296

Supervisor: Dr. Mohammad Saleh

2023

This project report is submitted to the Department of Computer Science and Engineering of Qatar University in partial fulfillment of the requirements of the Senior Project course.

Declaration

This report has not been submitted for any other degree at this or any other University. It is solely our work except where cited in the text or the Acknowledgements page. It describes work carried out by us for the senior project. We are aware of the university policy on plagiarism and the associated penalties, and we declare that this report is the product of our own work.

Student: Fatima AL Suwaidi Date: 18 – 05 – 2023

Signature:

Student: Fatma Jamal Date: 18 – 05 – 2023

Signature:

Student: Fatima AL Kuwari Date: 18 – 05 – 2023

Signature:

Abstract

In recent years, the number of students and faculty has drastically increased in Qatar University, especially in the College of Engineering. This increase comes as an advantage as a whole; however, it comes with many challenges, especially concerning the laboratories and the equipment inside. Many lab items that are borrowed by either faculty or students end up being lost, while others who are in need for these items might face difficulties finding items when needed. This problem includes many losses, including monetary, time and emotional loss. The proposed solution to this challenge is to have an efficient application that can successfully track every item in all laboratories in the university, whether it has been borrowed or being used in an alternative lab. The new application will be able to track the person who has requested borrowing it, the date and time it has been borrowed and its whereabouts in the campus.

The application's functional requirements include the ability to request borrowing an item, booking laboratories on specific times, tracking an outbound equipment, and returned equipment. Also, each person will have a unique user ID (such as university ID) and login information. This whole project will require a precise barcode system on each of the items in the laboratory. This labeling system will make it very easy to track every item, as it will automatically and immediately reflect on the application once someone has borrowed it. Faculty can also pick and choose equipment beforehand to use them in classes, guaranteeing that all the required equipment will be available for their labs. Also, the ability for faculty and students to book a lab will be evident on the application, where no duplicate bookings can occur. The damaged item can therefore be reported by students or staff by the barcode on the item, which can reflect instantly on the application.

The final advantage to this new system is the ability for admin staff to track the inventory levels of all lab equipment and the ability for them to see damaged equipment when reported on the system. This will enable the university to always keep the correct stock of items and order new equipment with accurate numbers as they will see the overall inventory levels. Also, it will help the admin staff analyze which equipment have been mostly used by faculty and students and which equipment has not been used. Admin staff can also create new items (for new equipment coming in), view item details such as suppliers and cost per item and finally, edit and archive the items.

Acknowledgment

We would like to show our appreciation and thankfulness to Dr. Mohammed Saleh for all the support and motivation given to us in the senior design project. He answered all our queries and provided great advice whenever we needed direction. Also, a special thank you to Engineer Mr. Tamer Ahmed for the support throughout the project, helping us improve our ideas and encouraging innovation and hard work.

Table of Contents

List of Figures

List of tables

1. Introduction and Motivation

The main problem that our project is solving relates to the laboratory equipment and laboratory bookings in the university. Problems arise when equipment is taken from the labs and stores, and their whereabouts and who borrowed them are unknown. This creates a hassle for both faculty and students. Additionally, the administrative employees in the university are unable to keep an accurate record of the stock of items and equipment, meaning items may be damaged, lost, or borrowed for a prolonged period of time without anyone noticing.

Also, many faculty may need to book laboratories in advance. The problem occurs when there is no immediately accurate system to track these bookings, edit them or reschedule them. It is challenging because many stakeholders are involved, including students, faculty, and admin staff in the university. The challenge here is to incorporate a barcode system for all items in the labs and all incoming new items in order to track them successfully. Once this is done, the whole project will surely be a success for years to come.

Our solution is to have a comprehensive, user-friendly application that allows students and faculty to book labs, request borrowing equipment, and for staff to easily track multiple borrowed equipment and the inventory levels of each item. This project aims to have a 'logistics' system incorporated to help everyone in the university who uses laboratories for any reason.

1.1. Problem statement

The problem currently is the inability for faculty and students to borrow equipment and make it trackable. Damaged, lost, and unreturned items might never be recorded and lost forever, causing inventory, and tracking to be invalid. The technical challenge may be to create the application with the correct coding and incorporate the barcode system successfully since each and every item must be labeled. This can be classified as a complex engineering problem since the barcode must instantly be reflected on the application in real-time. This also corresponds to the immediate computing function of the master data page, which shows current inventory levels and real time bookings. If there is a mass booking occurring (many people using the application at the same time), there needs to be no error that could be considered as a big problem, such as a double booking of a lab or supplicate item borrowing requests. The stakeholders here are affected negatively, from faculty, admin staff and the students themselves. This challenge might take time since there might be hundreds of items in the labs and storage facilities. Non-technical challenges might be training and getting all users to understand the application and how to use it. There might be a need for a user guide to explain in detail how to use the application. Our team aims to make it as user friendly as possible.

1.2. Project objectives

Old fashioned, manual methods of booking equipment and laboratories still exist today. We have seen that many departments in the university independently track information, many of them using manual excel spreadsheets and taking requests manually by phone or email. In order to create our user friendly, centralized application that gives everyone one single platform to perform all of these tasks and obtain data instantaneously, we have created following objective checklist:

- To create a user-friendly application that can be used by faculty and students.
- To cancel the need for manual inputs into spreadsheets and placing requests by phone or email.
- Incorporate a bar code system that labels every piece of equipment available in laboratories across the campus.
- Being able to sort and analyze and labeled items and equipment in the laboratories.
- To list the laboratory items on the application with respective images of the equipment.
- To code and create the application and incorporate all categories on the application, including booking functions, borrowing requests and reporting damaged equipment.
- Give special access to authorized admin staff to access the tracking system, including what items were taken and by who, which items were not returned and the overall inventory levels.
- To code a notice function, alarming admin staff that a certain low inventory level threshold has been reached.
- To be able to view instant data showing lab equipment and their location.

The success of the project will be judged when the user can access the application, book a specific lab at a desired time. On the borrowing side, the user must easily be able to request to borrow an item (from a clear drop-down menu of items) and the number of items that are to be borrowed (quantity). In the second phase of our project in the spring, we will definitely look at each individual objective above and measure the attainment in each one.

1.3. Expected benefits and impacts on various contexts

If there is no solution to the problem we are solving, there will be an impact on the learning of students if there is a shortage of equipment and items. If inventories are at low levels and nobody can track this, it may take a long time for suppliers to restock these shortages. Also, students and faculty might now be able to borrow equipment since they do not know how or where to go to do this.

Stakeholders will benefit since there will be an easy and straightforward way to book everything, especially in the palm of the hands as the phone is the most used source of technology in today’s world. The anticipated benefit in Qatar is the ability to incorporate an accurate logistics system in the laboratories. This can also help other departments in the future track their items. This is not limited to laboratories as it can expand to all sorts of materials including cleaning equipment, books, computers and many more examples. This system can also help other universities owned by Qatar who require such a tracking and booking. Borrowing system. A major example of courier services like Fedex or UPS, who have mastered this booking and tracking systems to save millions of dollars a day and to keep the stakeholders happy with the speed of reporting and tracking.

What got us interested in this project was when we saw that not all equipment in labeled correctly and that there are no was for us students or faculties to place a booking/borrowing request for any items. This project will help our coding skills and the ability to use similar systems in our future careers, whether in our future jobs or businesses. Additionally, the rapid growth of global technologies, interfaces, and software, means we are always attempting to adapt to these changes. We need to keep up with the global software and technologies by implementing new ideas and using innovation to keep up to date.

Table 1. Expected benefits and impacts on various contexts.

Context	Expected benefits and impacts
Individuals	Adaptation to the technologies available in the world today.
	Faster and more efficient use of the services without the need of manual/excel/physical booking systems.
	Admin staff are able to accurately track real time inventories and bookings on a master data page.
Organizations	Provide organizations like Qatar University and potentially others the ability to track inventory levels and also use this application for other items all around the university (such as cleaning equipment, food/beverage in the cafeterias) and much more.
	Organizations now have an accurate system that provides less made-made errors
	Track data on a large scale across the whole university
Society	This will provide adaptation to new technologies for the society. It is a big change once a society moves from a physical to an online platform in all aspect. This is similar to what is happening today from cash to the use of credit cards.
	This is linked to human well-being since the project will make it easier for all stakeholders to book, borrow items instantly wherever they are located.
	This is a "good" cultural change for the community as it provides everyone with more knowledge in the use of applications and virtual solutions.
Global	Applications like our one can have a great global impact as well as local since we can eventually patent it and sell it to other organizations around the world who may require digital solutions and tracking system software.
	Although we are not currently money-oriented, we hope that eventually we can get a patent and copyright on our application and sell it to corporations and institutions in Qatar and the world.

1.4. Market Research and Business Viability

Based on market research, there is a lack of a digital system that enables students to systematically share lab equipment. Most universities around the world require students to perform various tasks using the lab equipment and because there is limited supply of these equipment, all universities will need a system that allows booking of lab equipment. Hence, the market size can potentially include all universities around the world.

Additionally, it can be established that there is a need for a system that allows systematic tracking of lab equipment, prevents the user from keeping the item for longer than they actually need (through reminder alerts), and allows users to predict the availability of an equipment so they can set their plans accordingly.

The competing products will be the existing system used in the university, whether it is a manual spreadsheet that is filled by admin staff or the general software which is used by faculty to request lab bookings. The novel feature of the product is to combine all those features in a user-friendly and digital application, where booking, borrowing, lab requesting, and damaged equipment notices, etc. are made on a single platform.

The system can be enabled to be used by other universities based on a subscription system, per user. There can also be packages provided to universities based on their sizes and amount of usage and contracts to be agreed upon. The subscription can start from 5 USD per user and is subject to change based on infrastructure costs, such as servers for backend and databases. The product is to be free of charge for all students and faculty members for Qatar University, as it will be built in-house.

To bring our product to market, we will need to complete the prototype and all testing, then bring it forth to the university management for possible funding. This application might even be used in the future on a bigger scope, to include other inventories and booking systems, that might include, but not limited to, gym/sports venues, halls, other item inventories.

2. Background and related work

2.1. Background

This project aims to build a mobile application where it will help Qatar university and potentially many other universities in the country and region. Additionally, it will help students who are one of the main stakeholders by allowing them to engage in main functions of the application such as booking labs and equipment, reporting damaged items and tracking inventory levels for all labs on a single platform. This gives the admin staff the opportunity to conveniently and easily track inventory levels and know which labs are booked and by whom, all instantly on the application. The key problems with the current system are the inability to track any items' whereabouts and not being able to track which laboratory is booked. The app should be able to successfully solve this problem, leaving no room for low inventory levels, no duplicate bookings and most importantly, no paperwork or manual old-fashioned methods to book labs and equipment. Hopefully, the end result will help the university save valuable time and money by using this efficient, centralized portal.

2.2. Related work

After a thorough investigation on related work in Qatar and other regions, we have come to see that many websites contain features of the application that we are attempting to create. For example, in Qatar, the University of Doha for Science and Technology (UDST) has placed an online booking system for their digital media center items [1]. Their main features include drop-down menus with the specific items to be borrowed, a list of items with the borrowing price (and damaged contingency price), with an online consent signature for the end user or borrower. This shows how UDST has dealt with the problem at hand; however, it will not show any inventory levels or the names of the borrows and their locations, since this is confidential information that can only be accessed by authorized admin staff, as we are planning to use in our application.

On the other hand, applications such as Clustermarket have attempted to solve the problem by using the technical aspect, managing inventory levels and providing detailed reports on utilization [2]. Our application aims to combine those features of UDST mentioned above and Clustermarket, all on one platform. There are also publications related to our project by Mapua University [3], Gokongwei College of Engineering [4], which provide insight on how to create a laboratory borrowing system. Gokonwei came with a nice approach by installing a system which detects unvalidated equipment leaving the lab, similar to shops in malls. However, this will be more costly and not feasible in the university in the short term. Mapua University has provided a good insight on the project, however, uploaded manual forms which have to be scanned and re-uploaded for consent. Our project aims to have a complete online system that does not require paperwork. University uses all the components of the application from borrowing to returning but does not have a tracking system for administrators [7]. Perhaps they use a manual method for tracking, which does not suit our needs.

A publication for a very similar system to ours is available in Lorma College's [5]; however, it is not implemented yet. It uses a log-in system specific to the application. Our aim is to have the already university emails for logging in and requesting items. When it comes to the coding of the borrowing and tracking system, related works include those of [6,8]. Their code is heavily 'checkout' oriented, for businesses or universities who aim to make a profit from the lab equipment. Although the first few phases of their program are something we can learn from, their final 'checkout phase' is about payments and profit tracking.

Related work inside Qatar is quite limited to UDST which we mentioned earlier and Qatar National Library [9]. They have a state of the art borrowing system, including automated emails for late returns. We can definitely think of a similar notice for our project in the future, in the planning phase. We are surprised, however that QNL required a manual form filling and return, which we do not recommend. Also, their inventory is solely books, while we require a bigger range of items. Qatar University also has a good library book borrowing system, which includes search bars, study room bookings and epayments [10].

As a conclusion, we need to incorporate all these systems into one, centralized, user-friendly platform which includes both lab bookings and lab items borrow requests. These two functions of our application can then be tracked by admin staff, looking at inventory levels as well. There are many great applications and similar work however, each one seems to concentrate on a specific function and not all combined on a single platform, which we will strive to do and solve this problem.

Table 2.Comparison Table

Systems Features	University of Doha for Science and Technology System (UDST)	Clustermarket	Gokongwei College of Engineering publication	Lorma Colleges publication	LabSpace (Our System)
Inventory Levels Monitoring	No	Yes	No	No	Yes
Detection of unvalidated equipment leaving perimeter	No	No	Yes	No	No
Manual Forms to be validated by lab related staff	Yes	No	Yes	No	No
Paperless	No	Yes	No	Yes	Yes
Tracking system for Administrators	No	Yes	No	No	Yes
User registration and	No	Yes	No	Yes	Yes

authentication support					
Business Module including online payments	No	No	No	Yes	No
Email Notification System	No	No	No	Yes	Yes

3. Requirements analysis

3.1. Software development process

For our project, we have chosen to use Scrum methodology from agile development model. Some common software development processes include:

- **Agile:** An iterative and incremental approach to software development that emphasizes flexibility, collaboration, and rapid delivery of working software.
- **Scrum:** An Agile framework for managing software development projects that focuses on delivering working software in short sprints.
- **Kanban:** A workflow management method that helps teams visualize and optimize their workflows to improve efficiency and reduce bottlenecks.
- **Waterfall:** A linear, sequential approach to software development where progress flows in a single direction, with each phase of the development process being completed before the next one begins.
- **Extreme Programming (XP):** An Agile methodology that emphasizes the values of communication, simplicity, feedback, and courage.
- **Test-Driven Development (TDD):** A software development process in which automated tests are written first, followed by the implementation of the code that passes those tests.
- **Continuous Integration (CI):** A software development practice that involves frequently integrating code changes into a shared repository, typically multiple times per day.
- **DevOps:** A set of practices and tools that bring together development and operations teams to improve collaboration, automate processes, and increase the speed and reliability of software delivery.

Note that, these are just a few examples and there are many other software development processes and methodologies that teams can use depending on their specific needs and goals. Additionally multiple development processes can be combined to achieve goals in the most efficient manner

Although there are a few software development processes, we found the Scrum methodology to be the most all rounded in comparison to the rest. Because our project is an application which requires us to constantly check for feedback and improvements/troubleshooting, the Scrum method works great for the successful outcome of the project.

Scrum's iterative and incremental approach allows for flexibility in the development process. Our team can prioritize the requirements and deliver the most important features first. This will allow us to deliver a minimum viable product (MVP) as soon as possible, and start testing to get feedback and validate, and then add new features and functionality in subsequent iterations. This approach is particularly useful for our project as it allows us to adapt to the changing requirements of the project based on important stakeholders in this project such as Lab Supervisors, Faculty, Heads of sections.

By working on iterations, this gives us the ability to split tasks amongst our team members, such as planning, coding, design, and testing. We will then share our findings and discuss improvements. Booking, borrowing, and reporting are the main features of the application, which is heavily reliant on customer or client inputs, making the agile method suitable for us, giving us a clear direction on how to move forward with our amendments and decisions. This will help us to adapt and improve our application at a fast rate, saving us time as long as we work in efficient teams and plan well.

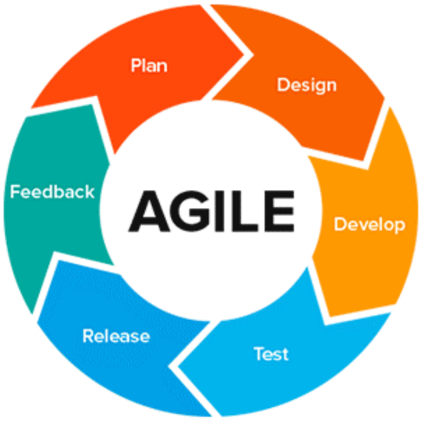


Figure 1.Agile Development Process which includes Scrum Methodology

3.1.1. Applying the software development process

We applied Scrum software development process by:

1. Developing a backlog for our project: creating a prioritized list of features, functions, and requirements for the software application, such as creating admin and user accounts, booking equipment, and managing equipment availability and listing them in Table 3: Use cases summary.
2. Sprint planning: breaking down the project backlog into smaller tasks that can be completed within a two-to four-week sprint and assigning each task to team members based on each member skill and availability as discussed in chapter 4.2: project timeline.
3. Weekly scrum: holding a weekly stand-up meeting to discuss progress, identify any issues, and plan the week's work.
4. Sprint review: holding a review meeting at the end of each sprint to demonstrate completed work to the supervisor and gather feedback.
5. Sprint retrospective: holding a retrospective meeting after the review to discuss what went well, what did not, and how to improve the process for the next sprint.
6. User opinion and thoughts: we developed a survey for Qatar University students to capture the needs and requirements of different types of students (all diagrams you will find them in appendix C – Survey Results).
7. Sprint backlog management: we manage the sprint backlog by adding, removing, or modifying tasks as needed to ensure that the team can complete them within the sprint.
8. User testing: we involved users in the testing of the software application process to ensure it meets the needs and expectations of different users such as admin, faculty, and students.
9. Continuous integration and deployment: we were continuously integrating new code changes into the application and deploying them to a staging environment for testing and feedback.

3.2. Functional requirements

The system shall act as common platform for both faculty and students at a University and allow them to perform tasks concerned to them. The students can borrow and return the items. The faculty can also perform some management activities. The Use Case diagram below shall map all the Use Cases to respective roles. Some use cases are not available to all roles as they are not authorized to perform those actions. Further description will be provided in the table following the diagram.

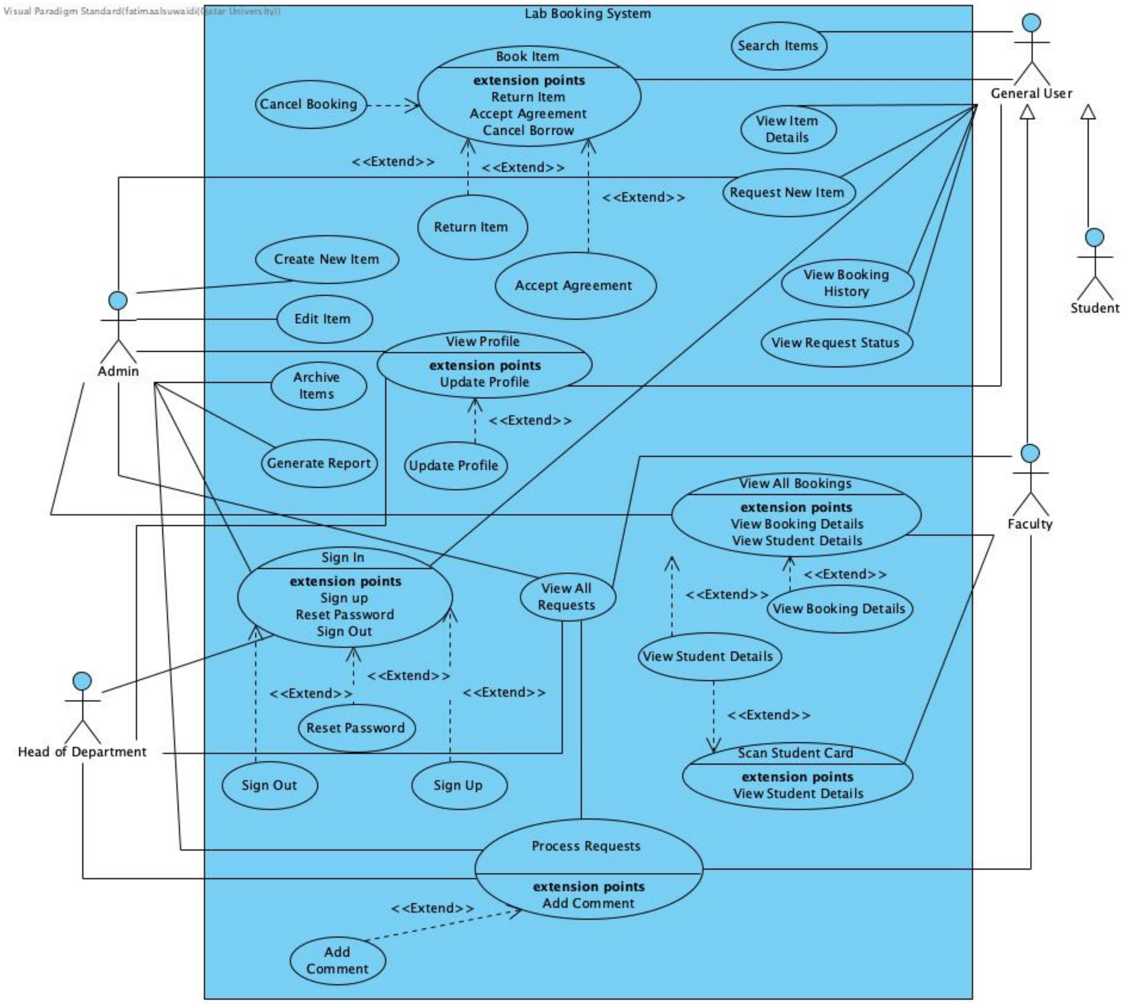


Figure 2.Use cases diagram.

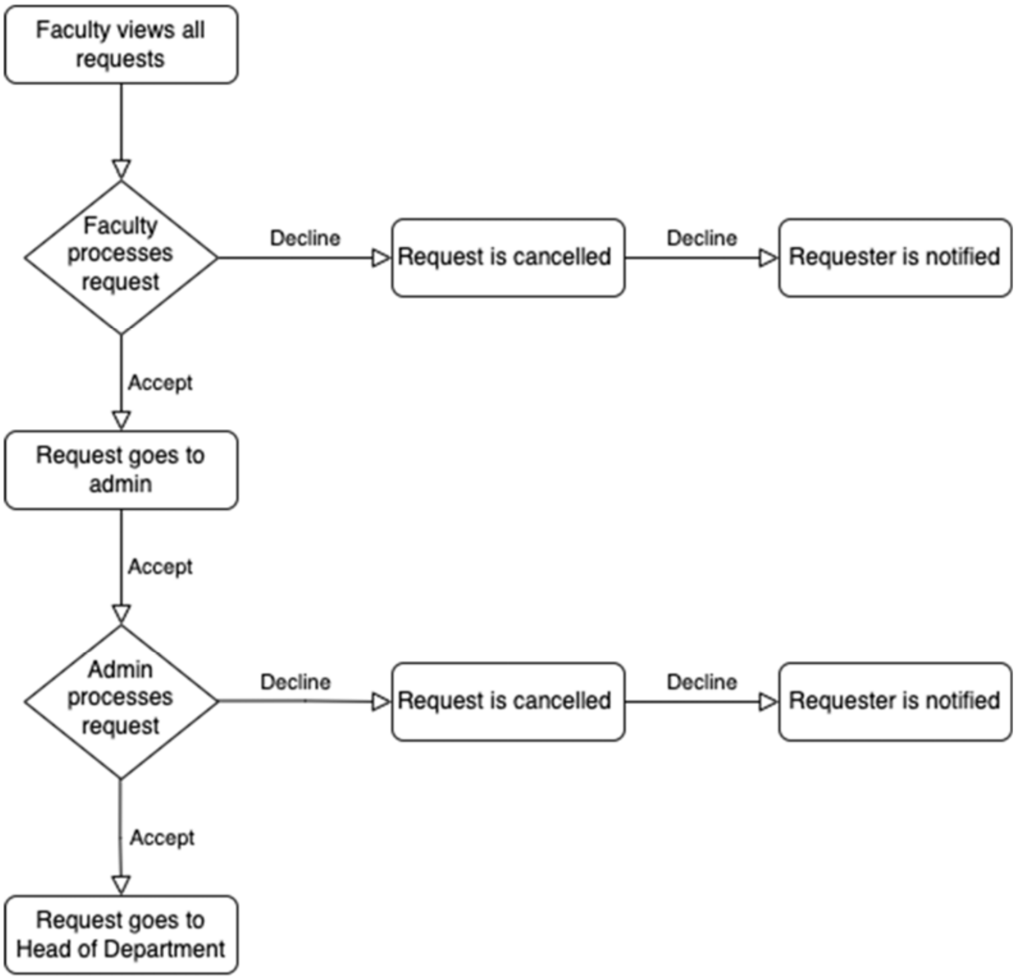


Figure 3.Flow chart for processing requests

Table 3.Use cases summary

ID	Use case	Brief description
1	Sign In	Users enter their email and password to sign into the application.
2	Sign Up	Users make an account by selecting a role.
3	Sign Out	Users sign out of the application.
4	Reset Password	Users reset forgotten password.
5	Process Request	User approves or declines a request for an item.
6	Add Comment	User can add a comment to a request for an item.
7	View All Bookings	View list of all bookings that are current or upcoming in future.
8	View Booking Details	View all information of the booking.
9	Scan Student Card	The faculty scans the student card to get all information.
10	View Student Details	View all information about the student.
11	View Request Status	User views status of requests already made by them.
12	View Booking History	User views history of items that were booked by them in the past.
13	Request New item	User request new item that is currently unavailable.
14	View item details	The user can view all the item details and wither if the item is available or not.
15	Search item	User searches for an item by name or ID.
16	Book an item	User faculty and student can book an item.
17	Accept Agreement	Users accept agreement while booking an item.
18	Return an item	User returns an item which was already bored.
19	Cancel booking	User can cancel the booking of equipment.
20	Create new item	User creates new items in the system by entering the relevant item details.
21	Edit item	User updates details of an item.
22	Archive items	User archives an item that is no longer in use.
23	Generate reports	View report of all the past booking transactions.

24	View profile	Users can view profiles to check their personal details such as name, ID, items booked by them, etc..
25	Update profile	User can update details in their profile
26	View all request	User can view all requests for items

3.3. Non-functional requirements

Table 4.Evaluation plan for the non-functional requirement

Non-functional requirement	Measurable targets	Evaluation Plan
1. Usability	The system should be easy to use with minimum number of steps required. For first time user it should not take more than 2 minutes to onboard themselves, register and start using the features of the system	Hold a session of User-Testing with a minimum of 50 end-users. Measure time taken to perform the primary scenarios of the system. Additionally, gather their feedback on their experience which can be used to make further UI/UX improvements.
2. Performance	The system should be responsive and loading times should be reasonably fast (not more than 10 seconds at once) and in background, wherever applicable.	Multiple tests shall be performed for each use-case's scenarios, under variety of circumstances. Different circumstances shall be simulated such as poor internet connection or a slow hardware device. Anything that crosses the maximum threshold of 10 seconds shall be highlighted and further optimizations performed, wherever applicable.

3. Modifiability	Based on evolving requirements, the system should be easily modifiable with minimal cost. Software should be architected and modelled by foreseeing possible future requirements as much as possible.	Simulate a change of requirement in an use-case, by implementing the modification in the system. By making sure the modification does not require changes in multiple places and, as a result, cost/effort is lower than implementing the feature from scratch, it can be ensured that the system is modifiable
4. **Security**	All personal student information and college related data must be secured and viewable only by the concerned users. Best practices shall be implemented to ensure security, such as, end-to-end encryption.	Firebase provides built-in security in the Authentication module. Appropriate rules will be set in the database to ensure access by only concerned users. Tests will be performed to verify other users cannot access unauthorized users.
e 5**. 5. Reliability** **5.**	System uptime to be ensured by keeping in place appropriate fail-over mechanisms and regular backup systems in case of a disaster.	System Health Checker to be implemented to ping the backend and perform an action at regular intervals. Any errors shall be highlighted by the Health Checker.

3.4. Assumptions

In order to have the best possible outcome of our project, there is a need to brainstorm the drawbacks and challenges. The following assumptions are the potential failure points which have to be monitored carefully throughout all phases of the project:

- The user must have access to a secure application where it is regularly monitored and checked for potential bugs and application errors.
- All users must have access to a reliable high speed internet connection to avoid data loss and requests not going through successfully.
- The implementation of the project will rely on a successful labeling system (using barcodes) to label every single piece of equipment. If we are unable to label all equipment, then the application cannot be successfully implemented.

3.5. Ethics

Table 5.Project-related Software Engineering Code of Ethics and Professional Responsibilities

IEEE Code of Ethics

Sec. No	Code	Usage and practice to address an identified ethical issue
7.8	To avoid unlawful conduct in professional activities, and to reject bribery in all its forms.	Project team members will complete the application by themselves without any help from outside.
7.8	To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest, and realistic in stating claims or	Team members will be very cooperative and help each other by commenting on each other's work to successfully create the application without any errors.
7.8	To hold paramount, the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable	The application will not misuse any user private information and will be highly secure to prevent leaking of data.

	development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment.	

ACM code of Ethics

Sec. No	Code	Usage and practice to address an identified ethical issue
1.5	Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.	Team members should respect each other's opinion and ideas, accept other members' suggestions, and take them into consideration.
2.1	Strive to achieve high quality in both the processes and products of professional work.	The newest tools and software will be used to create the application, to have a high-quality application.
1.6	Respect privacy.	Team members will use personal information for legitimate purposes and without violating the rights of users by using the minimum amount of personal information.

4. Project Plan

4.1. Project milestones

Table 6.Milestone of the project

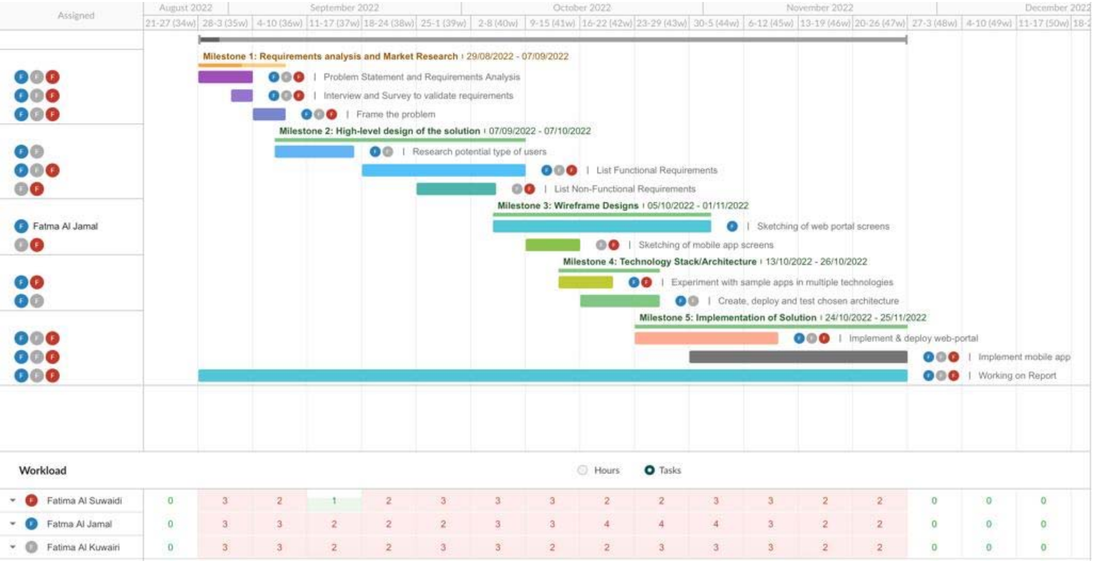
Milestone	Description	Deliverables
1- Requirements analysis and Market Research	Firstly, we had a brainstorming session with our supervisor to discuss different ideas and search about the concepts that we had in our mind to check whether the concepts are implemented or used in Qatar. After we looked over and investigated the concepts, we agreed that LabSpace will serve the needs in Qatar University. We met with Engineer Tamer to get ideas and more information about the labs and what we can add in our project to help the students and other faculty in the university. This meeting was helpful to us in that, he explained to us what issues they faced and what we can do to improve these issues.	Requirements
2- High-level design of the solution	In this milestone we had to list down the Functional Requirements that addressed the problems that were identified in the first milestone. While working on the Functional Requirements we had to identify the major stakeholders and decide how the system will connect everyone involved in the best way through different use cases. After listing down	Functional Requirements and Non-functional requirements

	the Functional Requirements we listed down the Non-Functional Requirements of the system keeping in mind the security and confidentiality of student data, user friendliness and performance of the system to allow pleasant usage experience for every kind of user: Admin, Student, Faculty, etc.	
3- Design of Wireframes	In this milestone we sketched the important screens and flows for the use cases we chose for Phase 1. We sketched multiple options for each screen and the flow from one screen to another, to decide on a design that could provide the best user experience.	Wireframes
4- Choose a Technology Stack	To address our solution, we had to choose a technology stack that was versatile and made the solution accessible to everyone. Additionally, we had to keep in mind the constraint we had of limited time and choose a stack that allowed us to maximize our productivity by reusing libraries and solutions already available on the web in the form of open-source software or other libraries. The above factors narrowed down our search for the best combination technologies to two major contenders: Firebase and Directus CMS. In this milestone we decided to go with Directus CMS and Flutter, after evaluating our constraints and requirements, which we listed in the previous milestone.	NA

5- Implementing the Solution	This milestone was a major stage and tricky one to finish on time, given the teams limited expertise in mobile and web applications. We had to divide the tasks in this milestone in a practical way to ensure continuous productivity of our team. We spent a significant portion of the time setting up the web portal and SQL database on a Linux server and developing the mobile application. After narrowing down the use cases to 25% of all the use cases for the first phase of this project, we began implementing the solution. We selected important use cases that would allow us to fully test our Proof-of-Concept solution to address the problems identified in the Milestone 1.	Proof of concept (web app and mobile app)

4.2. Project timeline

Senior1



Milestone 1: Requirements analysis and Market Research ׀ 29/08/2022 - 07/09/2022

| Problem Statement and Requirements Analysis

| Interview and Survey to validate requirements

| Frame the problem

Milestone 2: High-level design of the solution ׀ 07/09/2022 - 07/10/2022

| Research potential type of users

| List Functional Requirements

| List Non-Functional Requirements

Milestone 3: Wireframe Designs ׀ 05/10/2022 - 01/11/2022

| Sketching of web portal screens

| Sketching of mobile app screens

Milestone 4: Technology Stack/Architecture ׀ 13/10/2022 - 26/10/2022

| Experiment with sample apps in multiple technologies

| Create, deploy and test chosen architecture

Milestone 5: Implementation of Solution ׀ 24/10/2022 - 25/11/2022

| Implement & deploy web-portal

| Implement mobile app

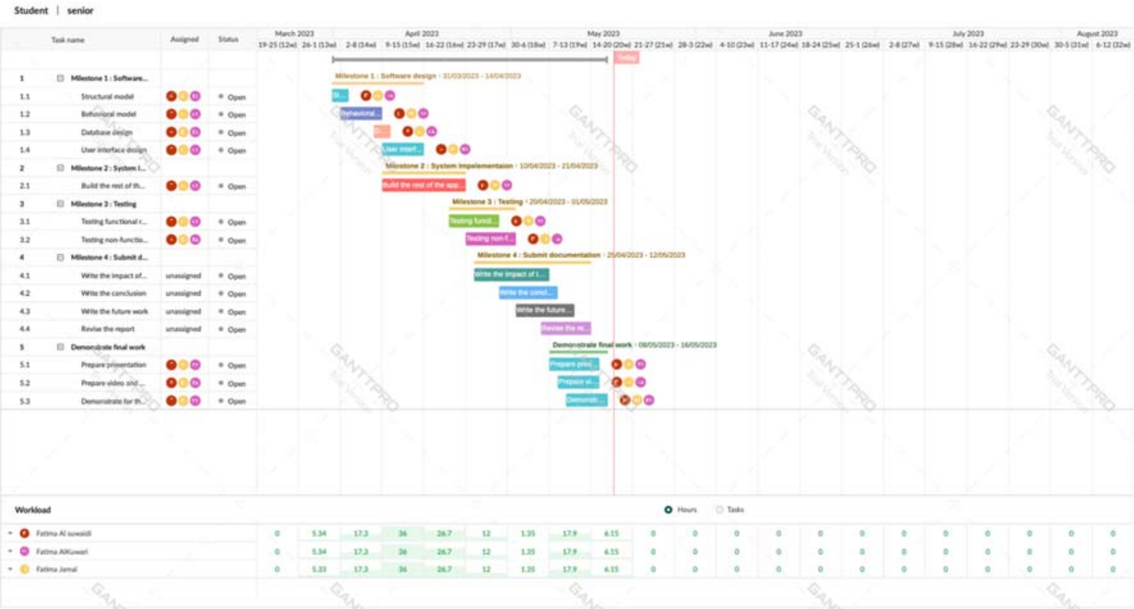
| Working on Report

Workload ○ Hours ● Tasks

Assigned	21-27 (34w)	28-3 (35w)	4-10 (36w)	11-17 (37w)	18-24 (38w)	25-1 (39w)	2-8 (40w)	9-15 (41w)	16-22 (42w)	23-29 (43w)	30-5 (44w)	6-12 (45w)	13-19 (46w)	20-26 (47w)	27-3 (48w)	4-10 (49w)	11-17 (50w)
Fatima Al Suwaidi	0	3	2	1	2	3	3	3	2	2	3	3	2	2	0	0	0
Fatma Al Jamal	0	3	3	2	2	2	3	3	4	4	4	3	2	2	0	0	0
Fatima Al Kuwairi	0	3	3	2	2	3	3	2	2	3	3	3	2	2	0	0	0

Figure 4.Gantt chart

Senior2



Student | senior

	Task name	Assigned	Status
1	Milestone 1 : Software...		
1.1	Structural model		Open
1.2	Behavioral model		Open
1.3	Database design		Open
1.4	User interface design		Open
2	Milestone 2 : System I...		
2.1	Build the rest of th...		Open
3	Milestone 3 : Testing		
3.1	Testing functional r...		Open
3.2	Testing non-functio...		Open
4	Milestone 4 : Submit d...		
4.1	Write the impact of...	unassigned	Open
4.2	Write the conclusion	unassigned	Open
4.3	Write the future work	unassigned	Open
4.4	Revise the report	unassigned	Open
5	Demonstrate final work		
5.1	Prepare presentation		Open
5.2	Prepare video and ...		Open
5.3	Demonstrate for th...		Open

Milestone 1 : Software design ׀ 31/03/2023 - 14/04/2023

Milestone 2 : System Impelementaion ׀ 10/04/2023 - 21/04/2023

Milestone 3 : Testing ׀ 20/04/2023 - 01/05/2023

Milestone 4 : Submit documentation ׀ 25/04/2023 - 12/05/2023

Demonstrate final work ׀ 08/05/2023 - 16/05/2023

Workload ● Hours ○ Tasks

	19-25 (12w)	26-1 (13w)	2-8 (14w)	9-15 (15w)	16-22 (16w)	23-29 (17w)	30-6 (18w)	7-13 (19w)	14-20 (20w)	21-27 (21w)	28-3 (22w)	4-10 (23w)	11-17 (24w)	18-24 (25w)	25-1 (26w)	2-8 (27w)	9-15 (28w)	16-22 (29w)	23-29 (30w)	30-5 (31w)	6-12 (32w)
Fatima Al suwaidi	0	5.34	17.3	36	26.7	12	1.35	17.9	6.15	0	0	0	0	0	0	0	0	0	0	0	0
Fatima AlKuwari	0	5.34	17.3	36	26.7	12	1.35	17.9	6.15	0	0	0	0	0	0	0	0	0	0	0	0
Fatima Jamal	0	5.33	17.3	36	26.7	12	1.35	17.9	6.15	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5. Gantt Chart

4.3. Anticipated risks

Table 7.Risks

Risk event	Approach to minimizing the effect on project success
Inability to correctly label hundreds of items	The barcodes must be double checked by admin staff before completing the labeling process. Each item must be registered on our master data sheet, and we must make sure it corresponds to the correct item physically.
Loss of master data	We must have a Google doc and GitHub with more than one source of backup in case information is lost at any specific point.
Not being able to create a user-friendly application	We must use the agile development process to check the user friendliness of the application. It would be a good idea to have potential end users test the application and we should incorporate all relevant feedback in multiple testing stages.
Application does not update information in real-time	We must make sure during out testing phases that information is updated in real time to avoid double and duplicate bookings on equipment, items, and labs. This might be a big problem if we do not test and find solutions to this risk.
No quick diagnostics if application suddenly crashes	Make sure that the technical help admin staff are aware on how to restart the application or provide almost instant help if the application crashes. If there is nobody to solve this problem when it happens, it may cause severe delays to the booking system and possible information loss.

5. Solution Design

We will discuss below 3 different software architectural structures to help us decide which one is more suitable for our project and do not have high risk that can affect our project.

Layered Architecture

It is one of the most popular architectural styles that organises the functionalities of similar modules or components into horizontal layers, each layer having a specific role in the application. There is no limit for the number of layers that the application can have to promote the concept of separation of concerns. It views the system as a whole and provides details to make the roles, responsibilities of the layers and the relationship between them understandable.

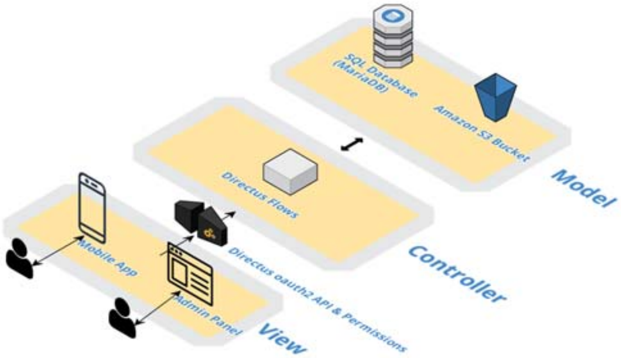


Figure 6. Layered Architecture

Pipe and filter

Pipe and filter is an architectural pattern for stream processing; it contains one or more components named filters. It filters the data and moves it through connectors named pipes. Filters can be viewed as functions such as sorting and counting and all filters can work together at the same time. When data is sent from the producer it passes through the pipes and filters until it reaches its destination.

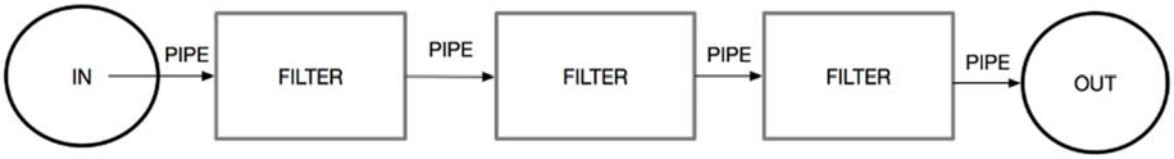


Figure 7.Pipe and filter

Object Oriented Systems

It is one of the most important concepts in developing software. It works by dividing responsibilities for an application system into reusable cooperative objects. Object Oriented architecture contains basic concepts shown in the diagram below:

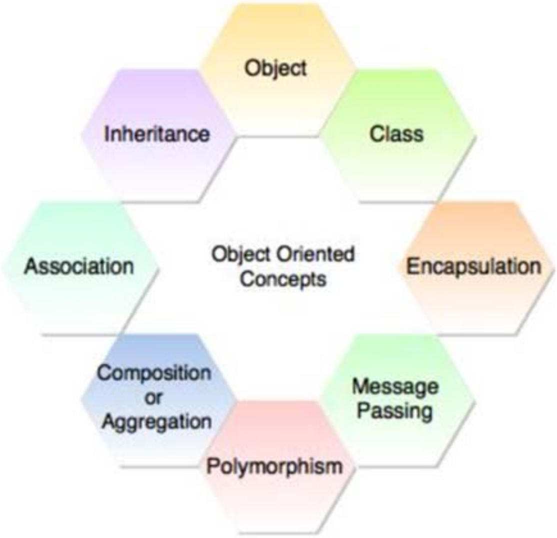


Figure 8.Object Oriented Systems

Table 8.Architecture Comparison

Architectures	Features	Drawbacks
Layered Architecture	Learning and implementing It is very easy and simple. The separation of function of each layer decreases the dependency. Testing the components individually will ease the testing part. The cost generally is suitable.	It may be insecure when bypassing layers is allowed. It may have high complexity in communication if it is not designed and anaged properly.

Pipe and Filter	Filters are reusable It makes sure of loose and flexible connections between components.	Filters may slow down when some pipe allow a single data type to pass through, which causes filters to do parsing. If a data is received and transformed by a filter as a whole such as an array, the data buffer may overflow. The Implementation of pipes and filters unknown.
Object Oriented Systems	It maps the application to real world objects to make it more understandable. It allows reusing the program by improving the quality of the system through object oriented architecture. It allows extending new functionalities without affecting the system.	Sometimes it does not lead to successful reuse on a large scale without an explicit reuse procedure. It is hard to have a limited time and budget because it offers new type of project management.

By comparing these three different software architectures we agreed that Layered Architecture is the most suitable to work on for our project.

5.1. High-level design

5.1.1. Alternative solutions and tradeoffs

Below, we will be discussing three alternative solutions for the requirements of our system. The first solution is a Mobile and Web-based Inventory Management System, which involves creating a mobile application and a web-based portal that allows students and lab supervisors to access and manage the inventory of equipment in the lab. The second solution is a Barcode Scanning System, which involves installing barcode scanners in each lab and creating a database of all equipment with their corresponding barcode. The third solution is an RFID Tracking System, which involves attaching RFID tags to all equipment and installing RFID readers in each lab. Each approach has its own advantages and disadvantages, which we will be discussing in detail below to weigh the solutions properly and decide the most cost effective, practical, and efficient approach.

1- Mobile and Web-based Equipment Management System

This solution would involve creating a mobile application and a web-based portal that allows students and lab supervisors to access and manage the inventory of equipment in the lab. The mobile application would allow students to view the inventory, request to borrow equipment, and check equipment in and out. The web-based portal would allow lab supervisors to update the inventory, approve or deny equipment requests, and generate reports on equipment usage and inventory levels. Both the mobile application and the web-based portal would be integrated to work seamlessly together, and they would be connected to a central database that would store all equipment and inventory information. Barcode scanning can be done through mobile cameras wherever applicable.

Advantages:

- Convenient for students and lab supervisors to access the inventory and manage equipment from their mobile devices and web-based portal.
- Automated notifications for students and lab supervisors.
- Reporting capabilities for equipment usage and inventory levels.
- Can be integrated with existing University's student and faculty database.

- Provides a centralized database to store all equipment and inventory information.
- Built-in mobile camera can be used to scan barcodes and save on hardware costs.

Disadvantages:

- May require significant development and implementation time and costs.
- Dependent on the internet connectivity and mobile devices' accessibility.
- Additional cost for mobile application development.

2- Barcode Scanning System

This solution would involve installing barcode scanners in each lab and creating a database of all equipment with their corresponding barcode. Lab supervisors and students would then use the scanners to check equipment in and out, and the system would automatically update the inventory. This solution would be particularly useful for labs with a large number of equipment, as it would make it easier to quickly check inventory and track equipment. This can also be integrated with a web-based system that allows students to book equipment before they come to the lab.

The barcode scanning system would be easy to implement and use, as most students and lab supervisors are already familiar with the technology. Additionally, it would be relatively inexpensive to implement as compared to other solutions, as it would only require the purchase of barcode scanners and the creation of a database. The system could also include features such as automatic email notifications for overdue equipment, which would help lab supervisors ensure that equipment is returned on time.

Advantages:

- Easy to implement and use, as most students and lab supervisors are already familiar with the technology.
- Relatively inexpensive to implement.
- Automated notifications for overdue equipment.

Disadvantages:

- May not be as durable as RFID tags.
- May not be able to read at a greater distance than RFID tags.
- Dependent on manual scanning for inventory check.

3- RFID Tracking System

This approach would involve attaching RFID tags to all equipment and installing RFID readers in each lab. The system would allow lab supervisors and students to check equipment in and out using RFID readers and would automatically update the inventory in real-time. This solution would be particularly useful for tracking equipment that is often moved between labs, as it would allow for quick and accurate inventory updates without the need for manual scanning. Additionally, students could be provided RFID tags as keychains or cards and further automation could be implemented.

An RFID tracking system would be more expensive to implement as compared to barcode scanning system, as RFID tags and readers can be more costly. However, RFID tags have a longer lifespan and are more durable than barcodes, which would make them a more cost-effective option in the long run. Additionally, RFID tags can be read at a greater distance than barcodes, which would make it easier to quickly scan equipment in large labs. The system can also have features such as automatic location tracking, which would allow lab supervisors to quickly locate equipment that is misplaced between multiple labs.

Advantages:

- Quick and accurate inventory updates without the need for manual scanning.
- Long lifespan and durability of RFID tags.
- Can be read at a greater distance than barcodes.
- Automatic location tracking feature.

Disadvantages:

- More expensive to implement than barcode scanning system.
- Dependent on RFID reader's range and battery life.
- Additional cost for RFID tags and readers.

It's important to note that tradeoffs and costs may vary depending on the specific implementations and number of departments covered in the University. It's also important to consider the long-term maintenance and scalability of the system, any constraints of the university and to consult with IT department and other stakeholders before choosing a solution. In our case we’ve chosen the first approach, which we further delve into, in the next section, considering the cost, time to implement and test, and the fact that smartphones and personal computers are readily available, unlike the hardware required in other solutions.

5.1.2. Selected solution overview

The selected solution for meeting requirements of our system is: Mobile and Web-based Equipment Management System, as it will be inexpensive to implement due to no requirement of new hardware. This solution involves creating a mobile application and a web-based portal that allows students and lab supervisors to access and manage the inventory of equipment in the lab. The mobile application will allow students to view whether an equipment is available, book an equipment for a certain period and make requests for equipment that are not available. The web-based portal will allow lab supervisors to update the inventory, approve or deny equipment requests, and generate reports on equipment usage and inventory levels. Both the mobile application and the web-based portal will be integrated to work seamlessly together, and they will be connected to a central database that will store all equipment and inventory information.

The main advantage of this solution is that it offers a convenient way for students and lab supervisors to access the inventory and manage equipment from their mobile devices and web-based portal. Automated notifications will keep students and lab supervisors updated on the status of their requests. The reporting capabilities will allow lab supervisors to generate reports on equipment usage and inventory levels, which will be useful for budgeting and forecasting future equipment needs. Additionally, the solution can be integrated with the existing University's student and faculty database, which means that students and faculty members can use their existing credentials to access the system.

Overall, the Mobile and Web-based Inventory Management System is a more comprehensive solution as compared to other alternatives which are more hardware-focused. It offers a centralized and digital way to keep track of all the equipment, which will improve the efficiency and accuracy of equipment management in the university labs.

An agile approach can be used for implementing the above where each member of the team can work on a specific component. This cross functional strategy will allow us to resolve the problem more efficiently and while constantly sharing our work with each other for feedback and advice.

5.1.3. High level architecture

One of the most popular architectural styles that was most suitable for our solution was Layered Architecture. The functionalities of similar modules or components can be placed into horizontal layers, each layer having a specific role in the application. There is no limit for the number of layers that a system can have, to promote the concept of separation of concerns. It views the system as a whole and provides details to make the roles, responsibilities of the layers and the relationship between them understandable. Below diagram shows a typical layer separation in a system.

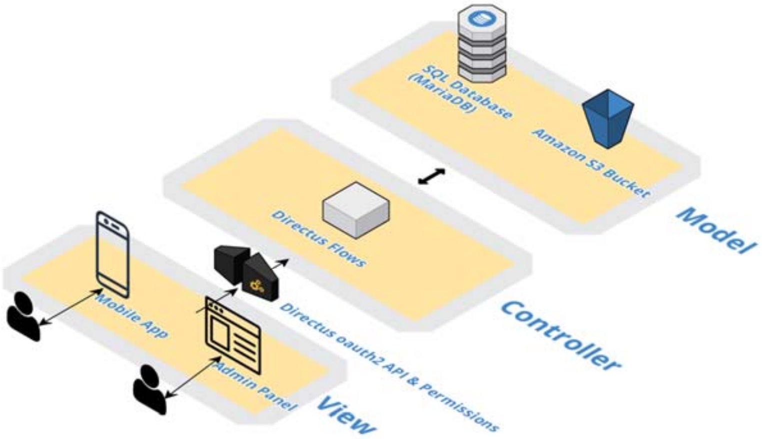


Figure 9. High Level Architecture Diagram

Any good software architecture should promote productive development by allowing development with minimal changes in multiple components, reusability, and testability. The arrangement of components must be tailored for the particular set of business requirements by accommodating both current and predicted future requirements. Keeping in mind the above, we have chosen to proceed with a layered approach.

For our system, we will separate the layers into four major concerns: Data, Model, Presentation/View, and Business Logic (Controller). One of the most popular architectures that separates the mentioned layers well is Model-View-Controller (MVC). To implement MVC architecture in Flutter we chose Getx state management system. Getx allows optimal separation of concerns by allowing navigation to be triggered from the Controller component, which typically cannot be done in Flutter. This separation of layers will also be used in the Web Application.

Model Layer will include all data models and repositories that will fetch data from the internet and transform them into usable models for our system to use. Repositories may make API calls to multiple sources to construct necessary models for the system. Pagination will be implemented for data of indefinite length to prevent long loading times and to comply with our non-functional requirements (see Table 3). For pagination to work events will be passed from other layers to trigger loading of new data and to communicate data when loaded.

View Layer will encapsulate everything related to Graphical User Interface such as colors, spacing, sizes, etc. It will communicate with the Model Layer through the Controller Layer, indirectly.

Controller Layer will abstract out all modifications to the State of the app through different functions, which will be called from the View Layer. View Layer will not directly modify any state.

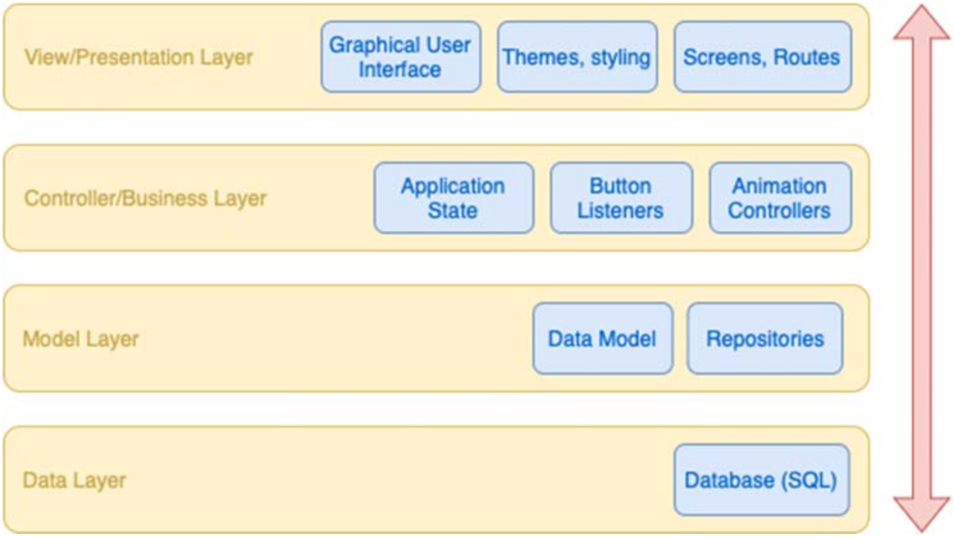


Figure 10.Layer Separation in LabSpace’s Mobile and Web App

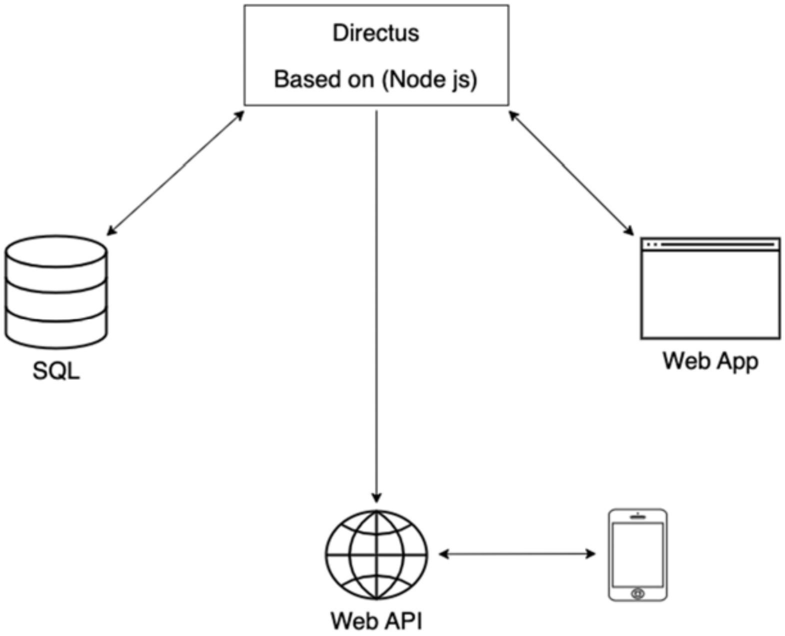


Figure 11. LabSpace’s High-level System Architecture

5.2. Structural model

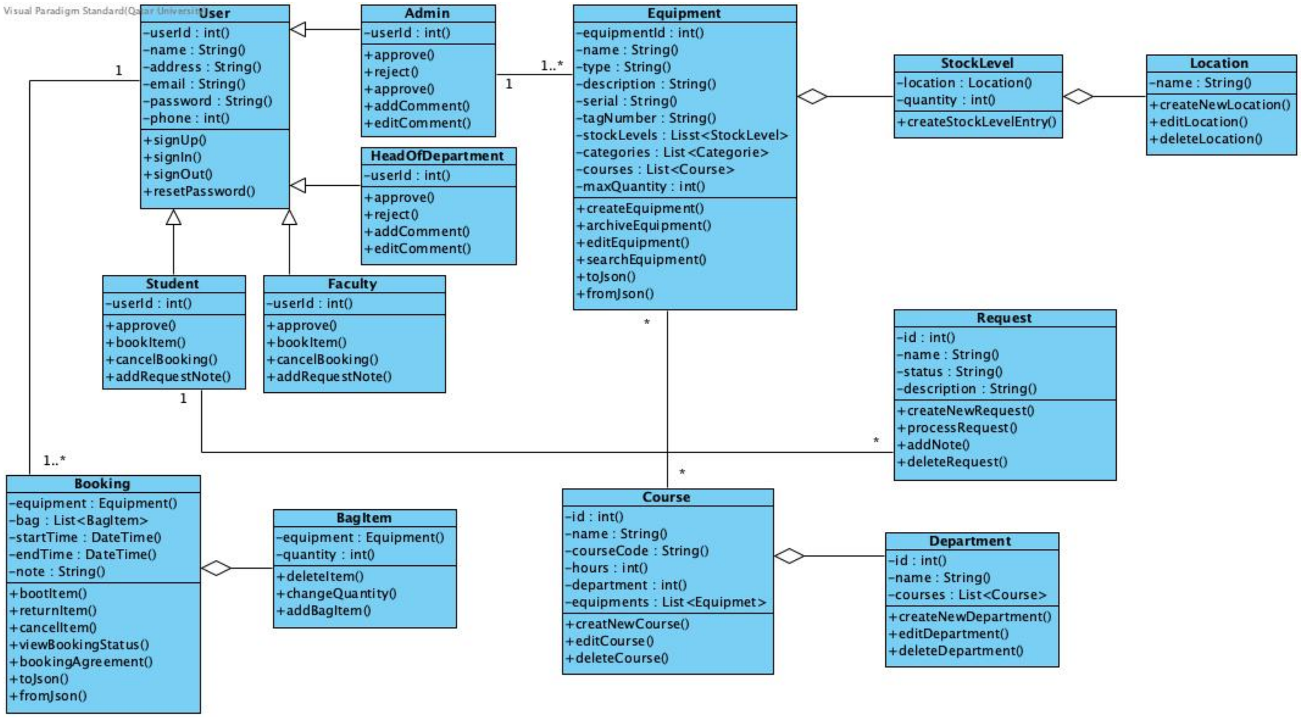


Figure 12. Class Diagram

This is our full class diagram for LabSpace app which consists of 13 classes, and they are: User, Admin, HeadOfDepartment, Faculty, Student, Booking, BagItem, Equipment, Course, StockLevel, Location, Request, Department.

As we can see we have directed associations between the classes (Example: Booking and User). Directed associations relates two classes together, it is identified as a solid line connecting two classes. The line may consist of one or more connected segments. The association can be shown between one-to-one, one-to-many, and many-to-many classes.

We also can see that we have a generalization (User, Admin, HeadOfDepartment, Student and Faculty). Generalization is a way of classifying similar objects in a more generalized manner. The entities show a parent-child relationship. This is represented as a hallow arrow shaped toward the parent class.

Furthermore, we have an aggregation (Booking has Bag item). Aggregations are also a type of association representing a two-way, highly dependent relationship between two different classes. If the composite is deleted, the other entities linked with it are also deleted. This is because the part highly dependent on the whole class and cannot be separated. This is represented as an empty diamond shape.

5.3. Behavioral model

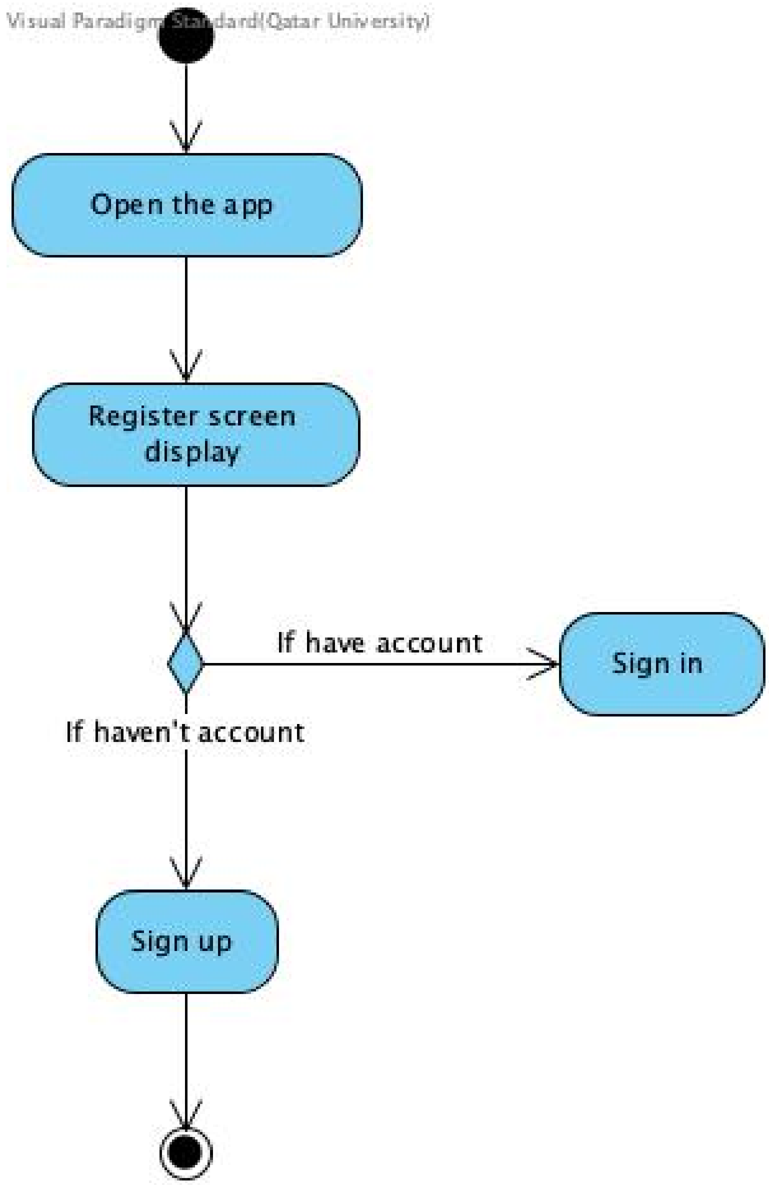


Figure 13. Registering Activity Diagram

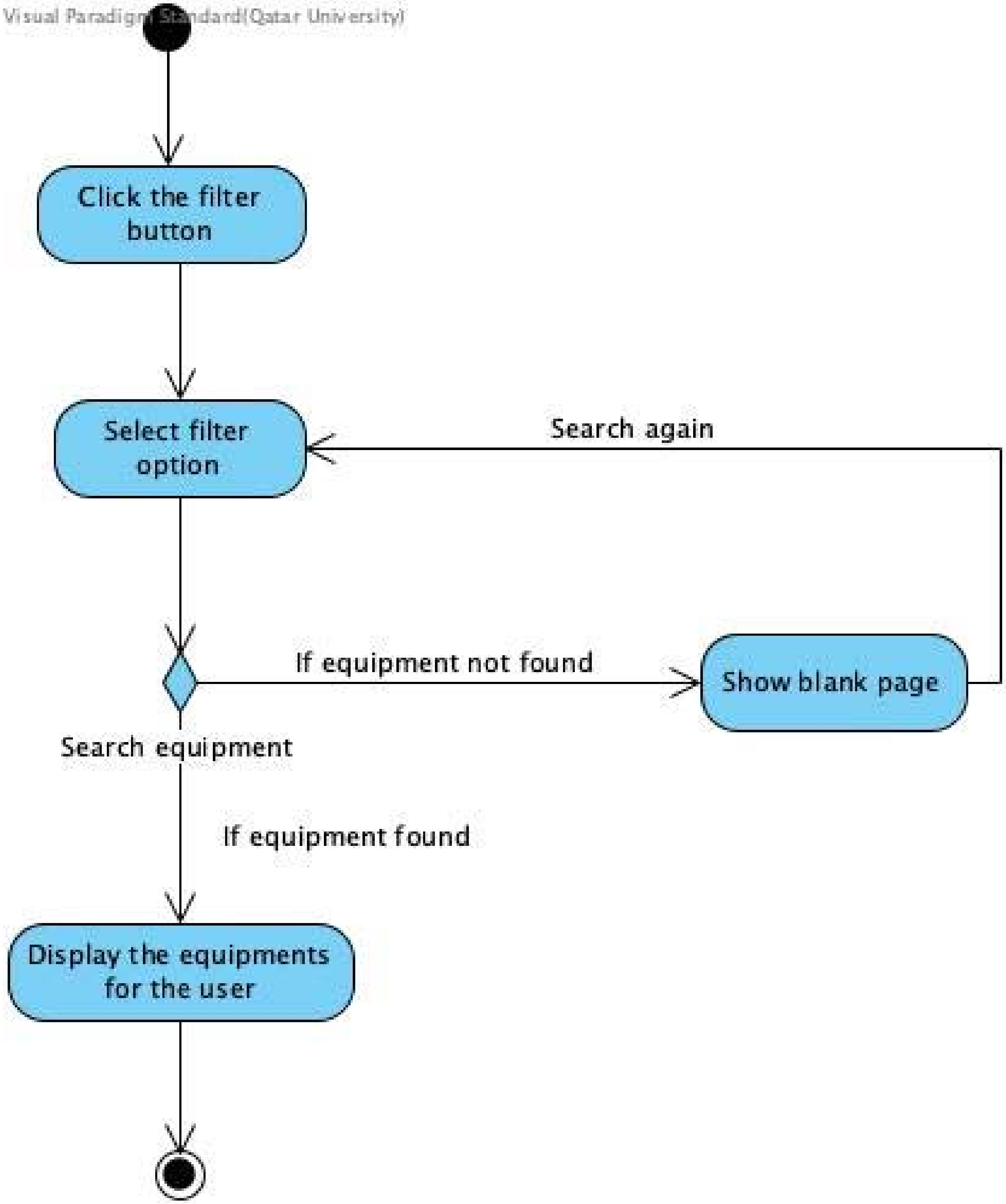


Figure 14. Filtering Properties Activity Diagram

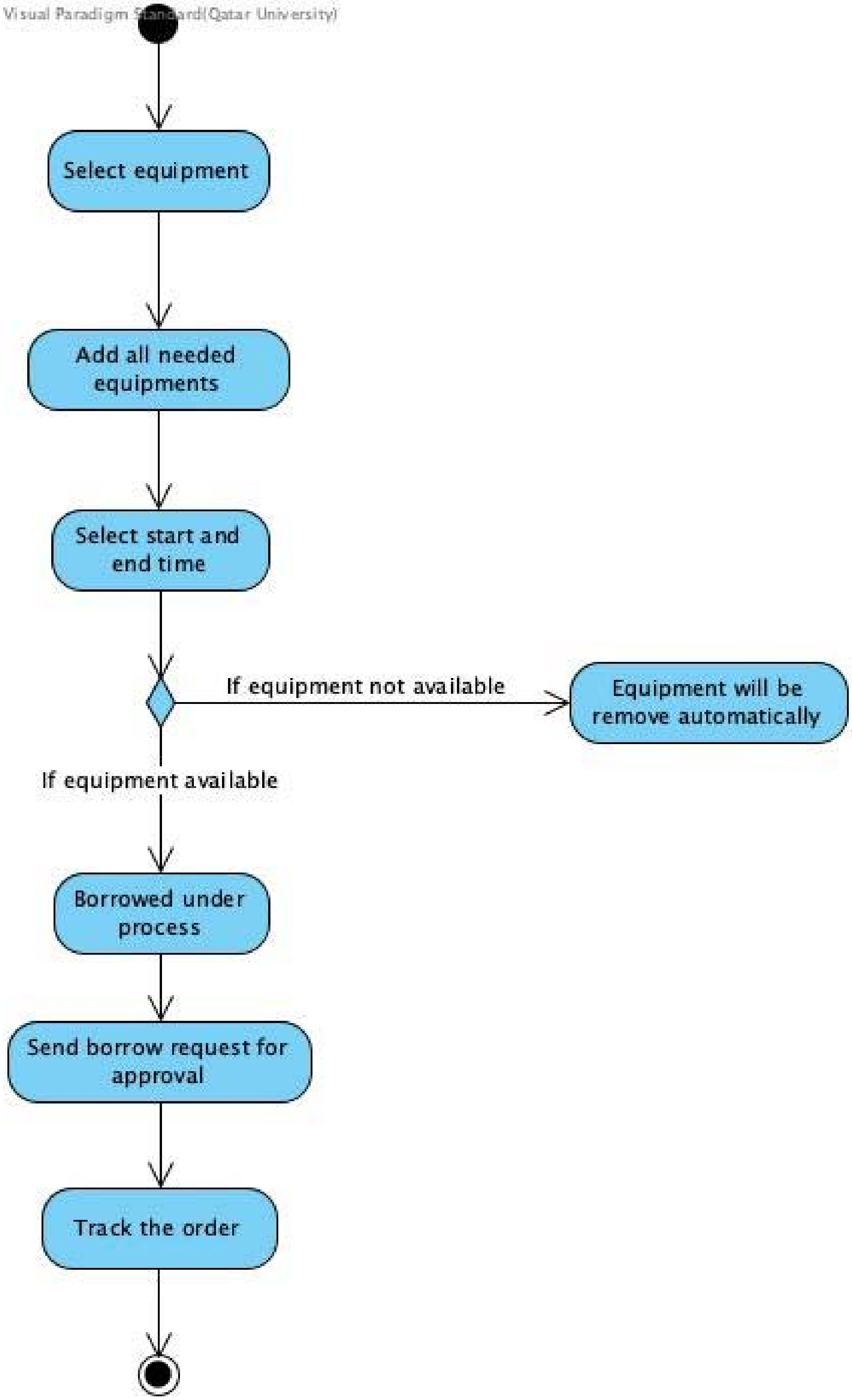

Visual Paradigm Standard(Qatar University)
Select equipment
Add all needed equipments
Select start and end time
If equipment not available
Equipment will be remove automatically
If equipment available
Borrowed under process
Send borrow request for approval
Track the order

Figure 15. Borrowing Activity Diagram

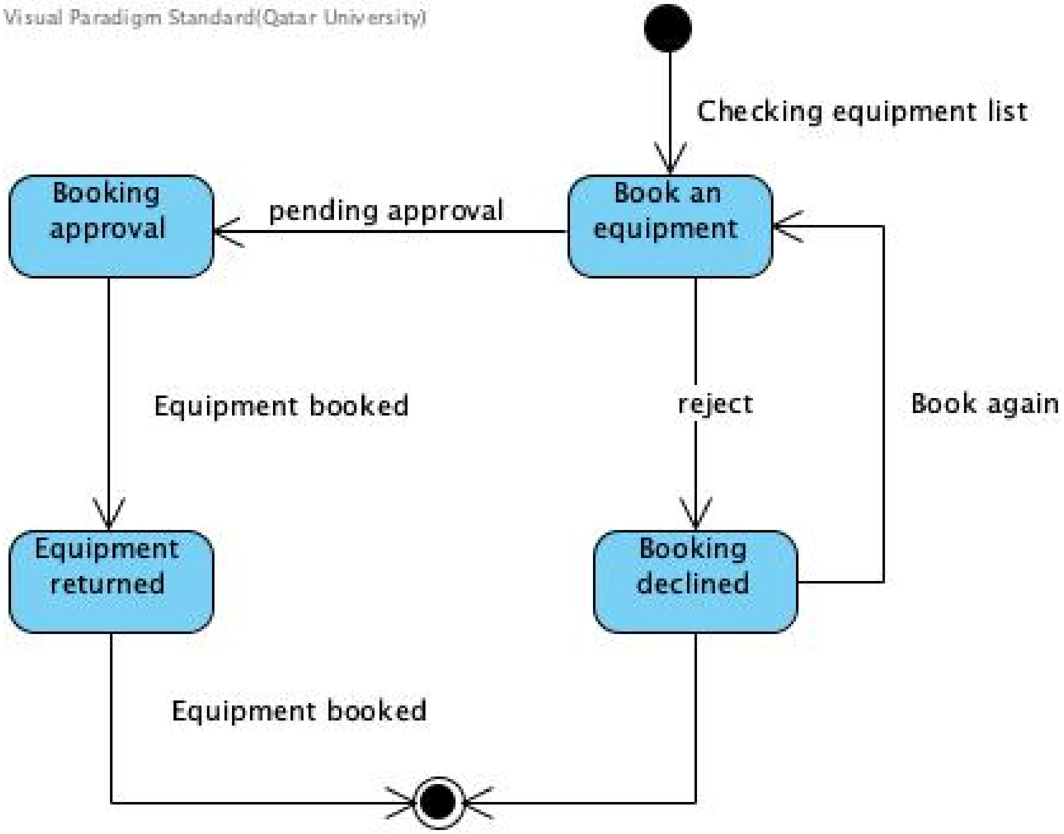


Figure 16.Booking Equipment State Diagram

Booking equipment state diagram is a visual representation of the process that a user goes through to book equipment in the application. It typically includes different states and transitions that show the various steps involved in the booking process. By using a state diagram, it becomes easier to understand the workflow of the system and identify any potential issues in the process. A well-designed booking equipment state diagram can help ensure a smooth and efficient equipment booking experience for users, leading to increased productivity and user satisfaction.

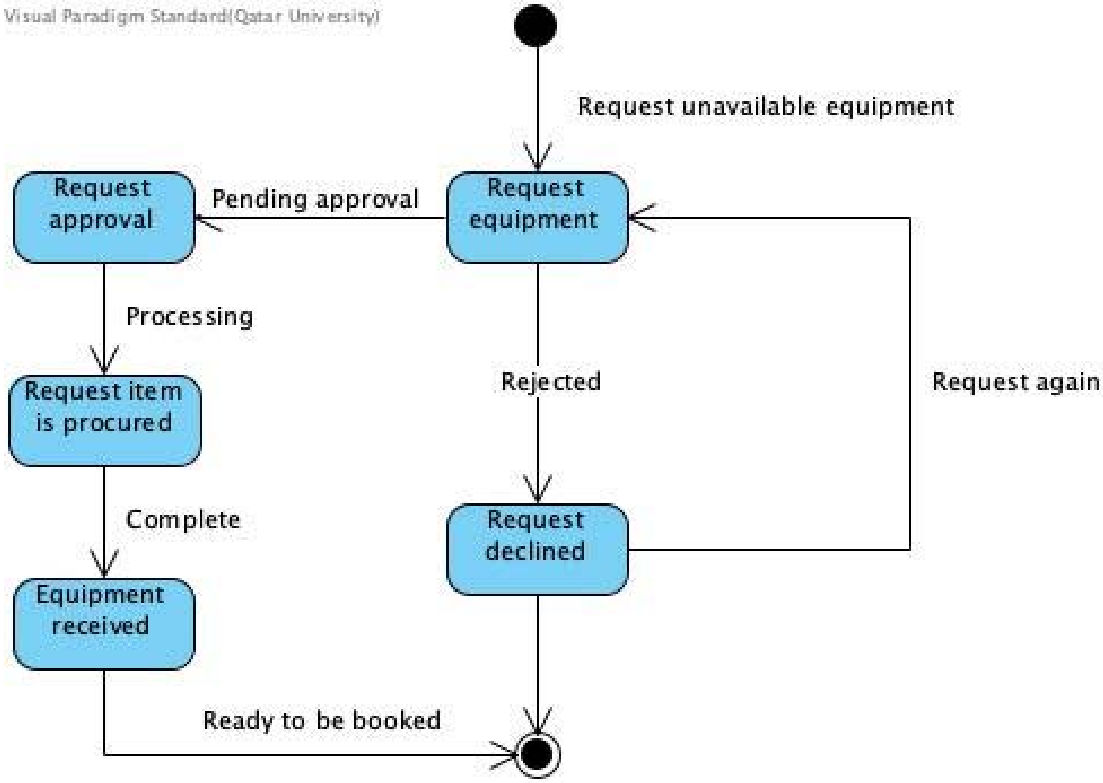


Figure 17.Request Equipment State Diagram

Requesting equipment state diagram is a visual representation of the process that a user goes through to request new equipment in the application. It includes different states and transitions that show various steps involved in the request process, such as providing a reason for the request, and submitting the request for approval. By using a state diagram, it becomes easier to understand the workflow of the system. The diagram can help lab administrators to track and manage equipment requests, making it easier to priorities requests and allocate resources effectively.

5.4. Database design

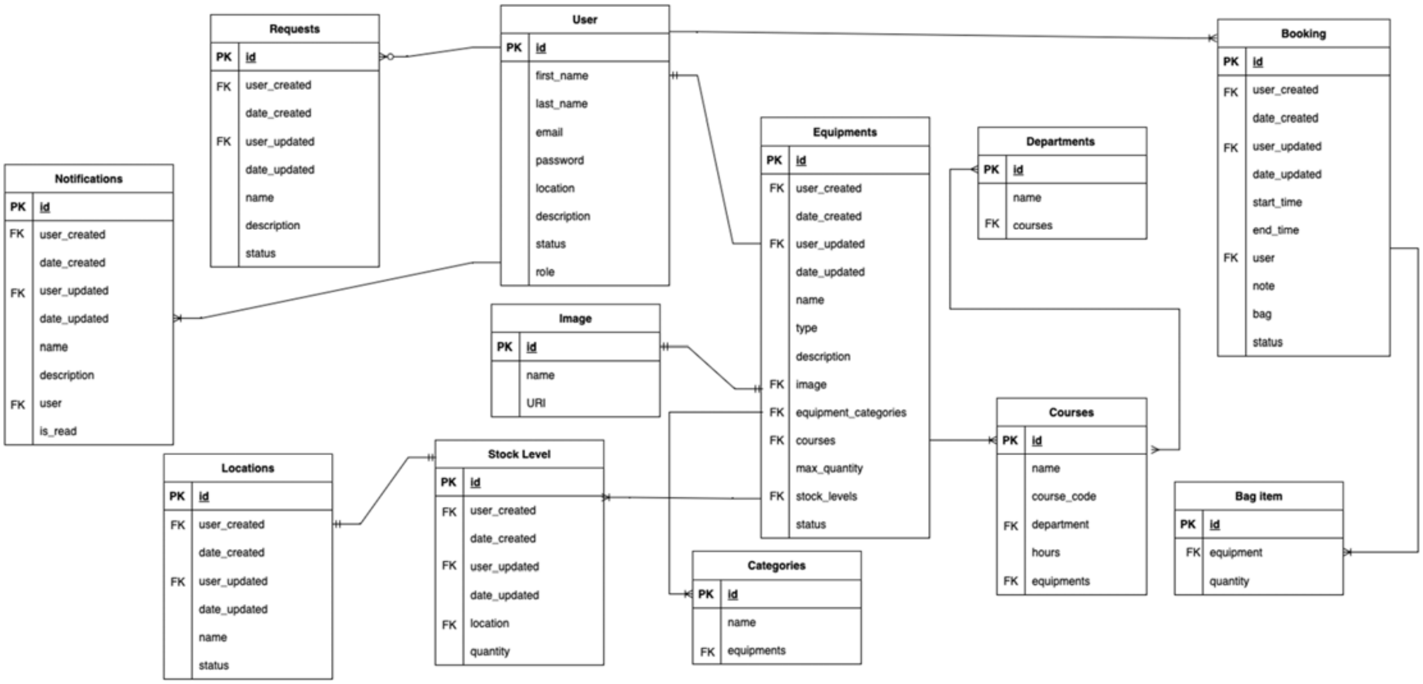


Figure 18. Entity Relationship Diagram of Database

Entity	Description
Equipments	This holds information about the equipment such as name, description, and image. This also holds relations (foreign keys) to other important tables such as Stock Level, Categories, Courses, etc. This separation of information and relationship structure is crucial to enable desired filtering and search capabilities in the system
User	To store all the user related information especially for authentication in the system and to manage permissions using “role” attribute, we use this entity
Stock Level	This contains location information and the quantity in each location. This is related to the Equipments entity to convey inventory levels of each equipment. One equipment can be related to many Stock Level entity as an equipment can be there in multiple locations

Booking	This entity holds details about duration of booking: Start Time and End Time and other related metadata. It's also contains list of references to BagItems entity which holds information about all equipments in the booking
BagItem	This entity stores quantity and relation to equipment. This tells how many units of equipment were booked. Limitation: This does not account for which location the equipment was taken from. A future phase of the system could accommodate equipment's location and serial number of each equipment too.
Notifications	This entity is related to each user of the system. Notifications entity holds standard information like message and time. This entity has only one relation, i.e. to the user.
Departments	This is displayed in the filters of the system and helps users filter courses by departments
Courses	The main purpose of this entity is to enable filtration of equipment in the system by courses. However, in future versions of the system the other metadata of courses can be utilized for advanced features.
Categories	This enables grouping of equipment into categories. Equipments can be part of multiple categories as well as a single categories contains multiple equipments, hence, the many-to-many relationship

5.5. User interface design

When you first launch the app, it will show you the sign in and sign-up option. Users that have an account can sign in directly otherwise sign-up. After it will take them to the home bag.

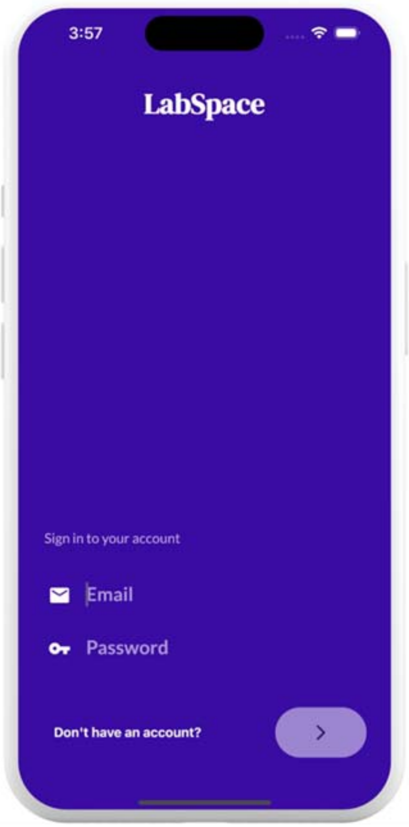


Figure 20. Sign-in

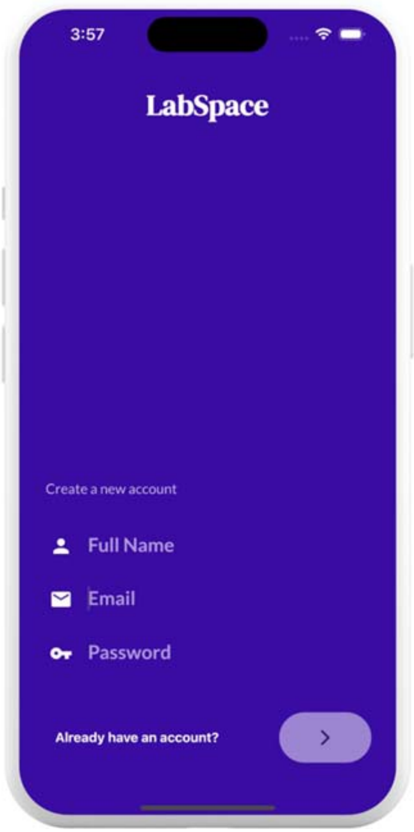


Figure 19. Sign-up

Sign in & Sign up:

The UI shows the username and password input fields, the user enters the inputs, and the authentication is saved in the database. Then the user can sign in with their credentials.

Edit profile:

User can navigate to the last tab, and there they can click the “profile photo” button and then edit their profile.

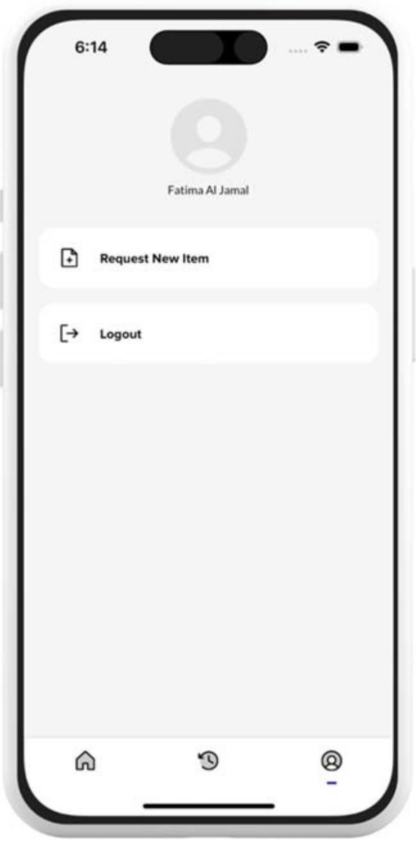
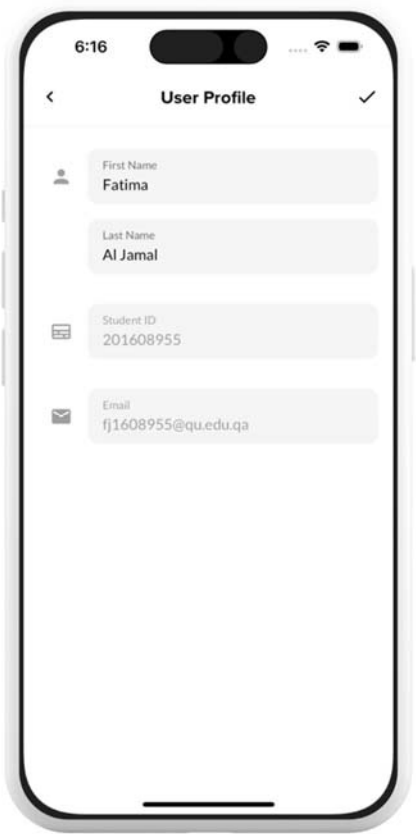


Figure 22. Profile page



First Name
Fatima

Last Name
Al Jamal

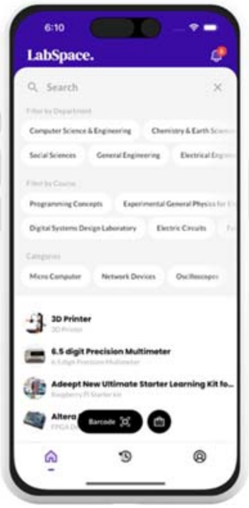
Student ID
201608955

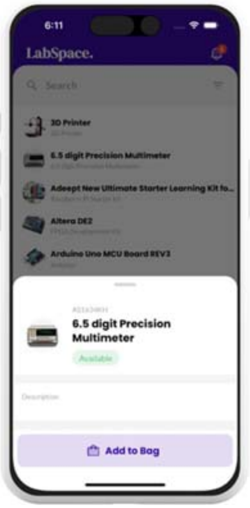
Email
fj1608955@qu.edu.qa

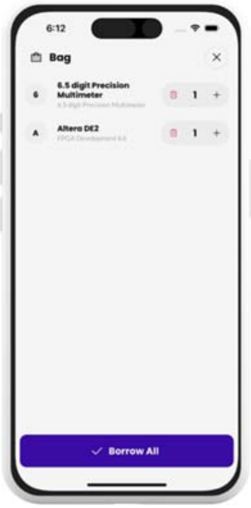
Figure 21. Edit profile page

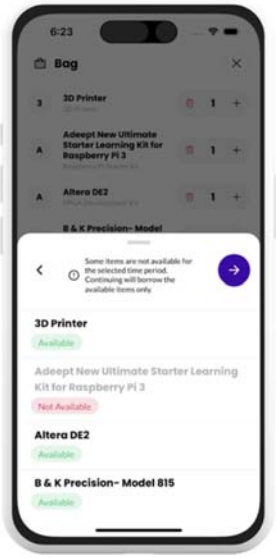
Borrowing equipment:

Firstly, the user can navigate to the first tab, then select needed equipment and add it to the bag. Secondly, the system shows whether the equipment is available or not. Then the user writes the purpose of the borrowing equipment and the agreement, the user can specify the period of borrowing “Start Time and End Time”. Thirdly, the user will wait for the admin reply if it is “Accepted or Rejected”. If the admin “Accepts” the borrowing request the user can borrow the equipment, and if the admin “Rejects” the borrowing request he needs to write the purpose of rejecting the borrowing request.



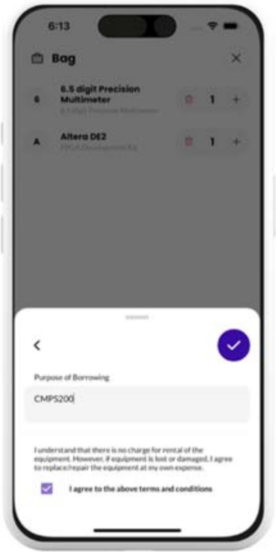

6:11
LabSpace.
Search
3D Printer
6.5 digit Precision Multimeter
Adeept New Ultimate Starter Learning Kit fo...
Altera DE2
Arduino Uno MCU Board REV3
6.5 digit Precision Multimeter
Available
Description
Add to Bag




6:23
Bag
3
3D Printer
1
A
Adeept New Ultimate Starter Learning Kit for Raspberry Pi 3
1
A
Altera DE2
1
B & K Precision- Model

Some items are not available for the selected time period. Continuing will borrow the available items only.

3D Printer
Available
Adeept New Ultimate Starter Learning Kit for Raspberry Pi 3
Not Available
Altera DE2
Available
B & K Precision- Model 815
Available

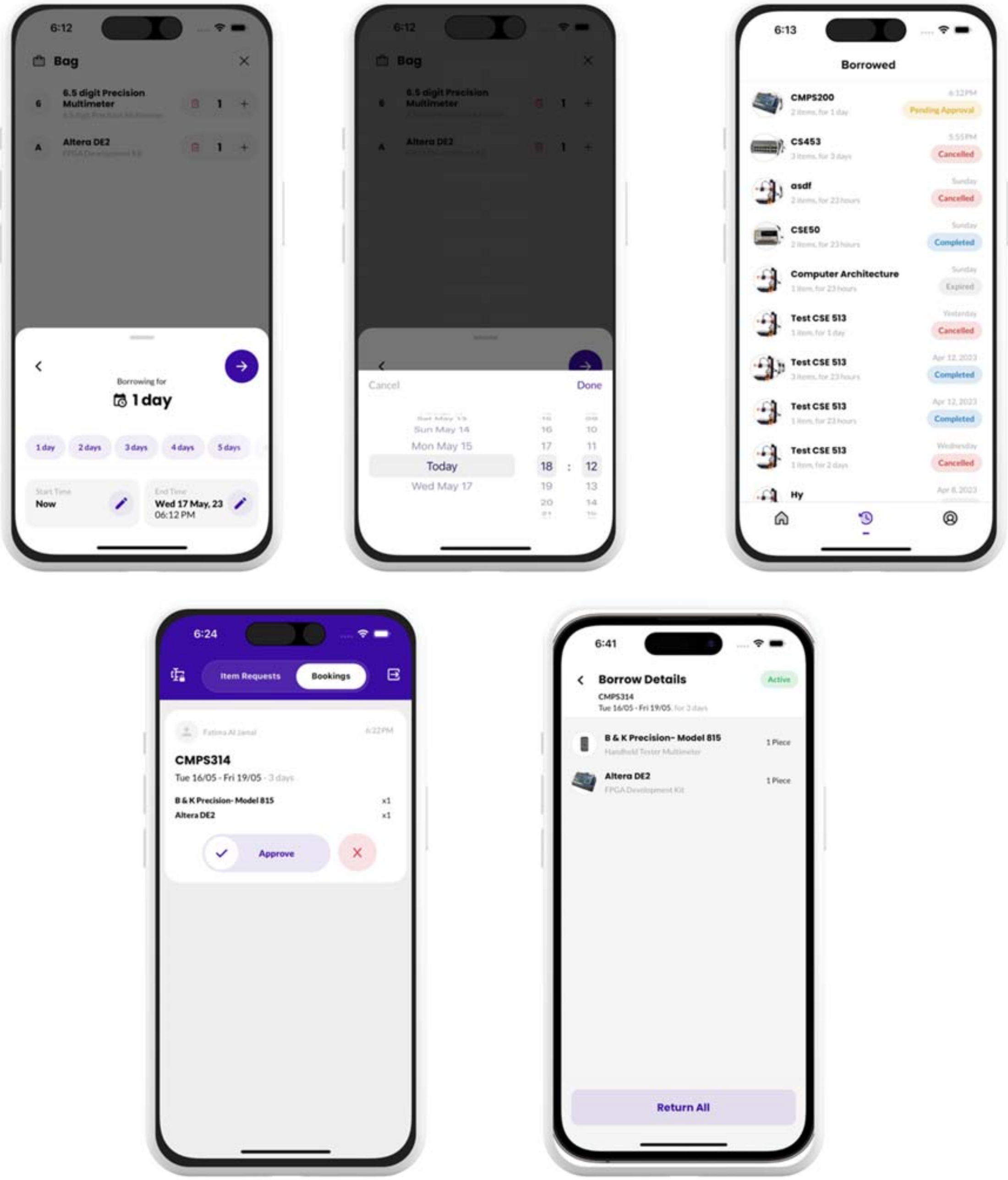


Purpose of Borrowing

CMPS200

I understand that there is no charge for rental of the equipment. However, if equipment is lost or damaged, I agree to replace/repair the equipment at my own expense.

☑ I agree to the above terms and conditions



6:12

Bag

6 6.5 digit Precision Multimeter — 6.5 digit Precision Multimeter — 1 +

A Altera DE2 — FPGA Development Kit — 1 +

Borrowing for

1 day

1 day | 2 days | 3 days | 4 days | 5 days

Start Time: Now

End Time: Wed 17 May, 23 06:12 PM

6:12

Cancel — Done

Sat May 13 | 15 | 09
Sun May 14 | 16 | 10
Mon May 15 | 17 | 11
Today | 18 : | 12
Wed May 17 | 19 | 13
 | 20 | 14
 | 21 | 15

6:13

Borrowed

CMPS200	2 items, for 1 day	6:12PM — Pending Approval
CS453	3 items, for 3 days	5:55PM — Cancelled
asdf	2 items, for 23 hours	Sunday — Cancelled
CSE50	2 items, for 23 hours	Sunday — Completed
Computer Architecture	1 item, for 23 hours	Sunday — Expired
Test CSE 513	1 item, for 1 day	Yesterday — Cancelled
Test CSE 513	3 items, for 23 hours	Apr 12, 2023 — Completed
Test CSE 513	1 item, for 23 hours	Apr 12, 2023 — Completed
Test CSE 513	1 item, for 2 days	Wednesday — Cancelled
Hy		Apr 8, 2023

6:24

Item Requests | Bookings

Fatima Al Jamal — 6:22PM

CMPS314

Tue 16/05 - Fri 19/05 - 3 days

B & K Precision- Model 815	x1
Altera DE2	x1

✓ Approve — ✕

6:41

Borrow Details — Active

CMPS314

Tue 16/05 - Fri 19/05, for 3 days

B & K Precision- Model 815 Handheld Tester Multimeter	1 Piece
Altera DE2 FPGA Development Kit	1 Piece

Return All

Figure 23. Borrowing equipment process

Cancel borrowing:

User can navigate to the middle tab “borrowed bar”, and there they click on the borrowed item that they want to cancel, click on the “cancel booking” button to cancel, then it will show in the top right it is cancelled.

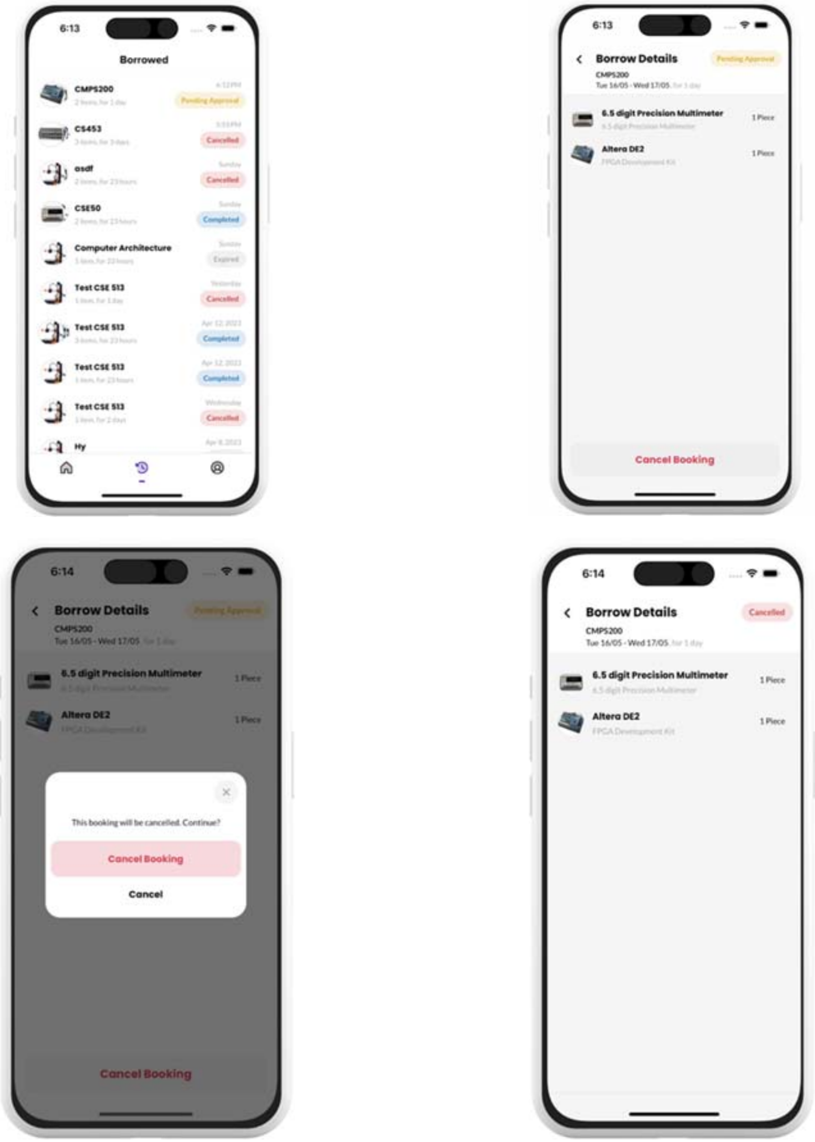
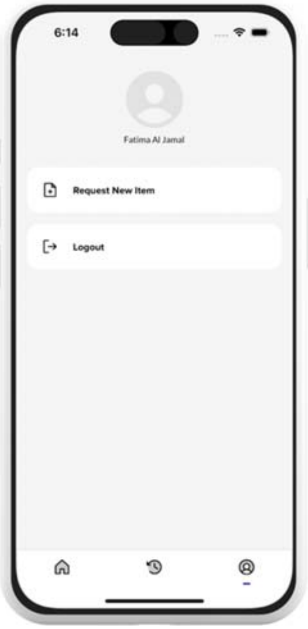
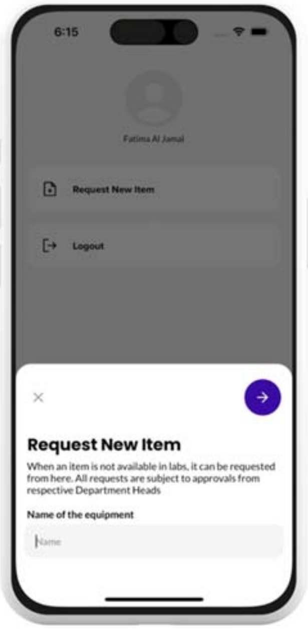


Figure 24. Canceling booking

Request new item:

User can navigate to the last tab then click the “Request new item” button, then add the name of the equipment that they want to request for, after that add a short description for the equipment. Finally, it will send a notification to the head of department and the admin to “Approve or Reject” it with the reason.

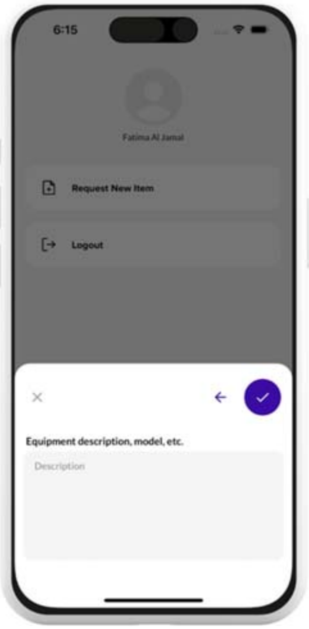




When an item is not available in labs, it can be requested from here. All requests are subject to approvals from respective Department Heads

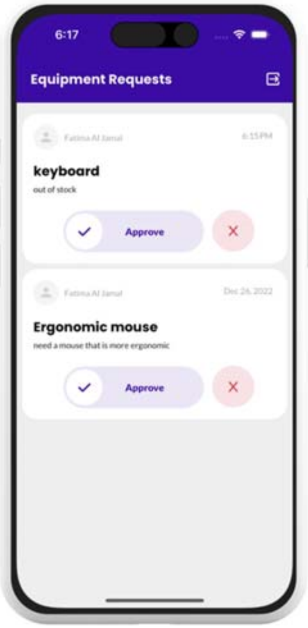
Name of the equipment

Name



Equipment description, model, etc.

Description



Fatima Al Jamal 6:15PM

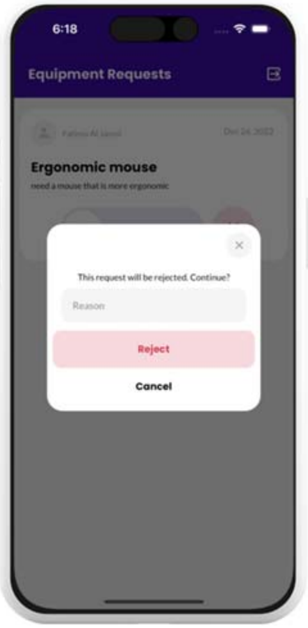
keyboard

out of stock

Fatima Al Jamal Dec 26, 2022

Ergonomic mouse

need a mouse that is more ergonomic



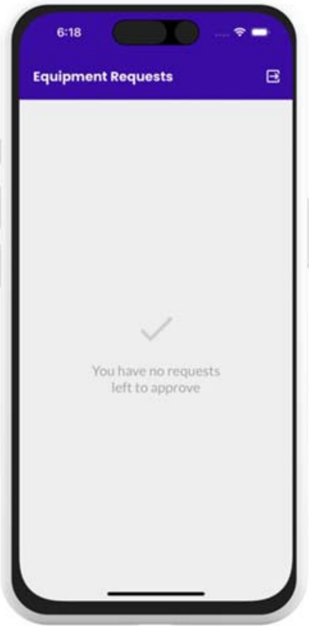
Fatima Al Jamal Dec 26, 2022

Ergonomic mouse

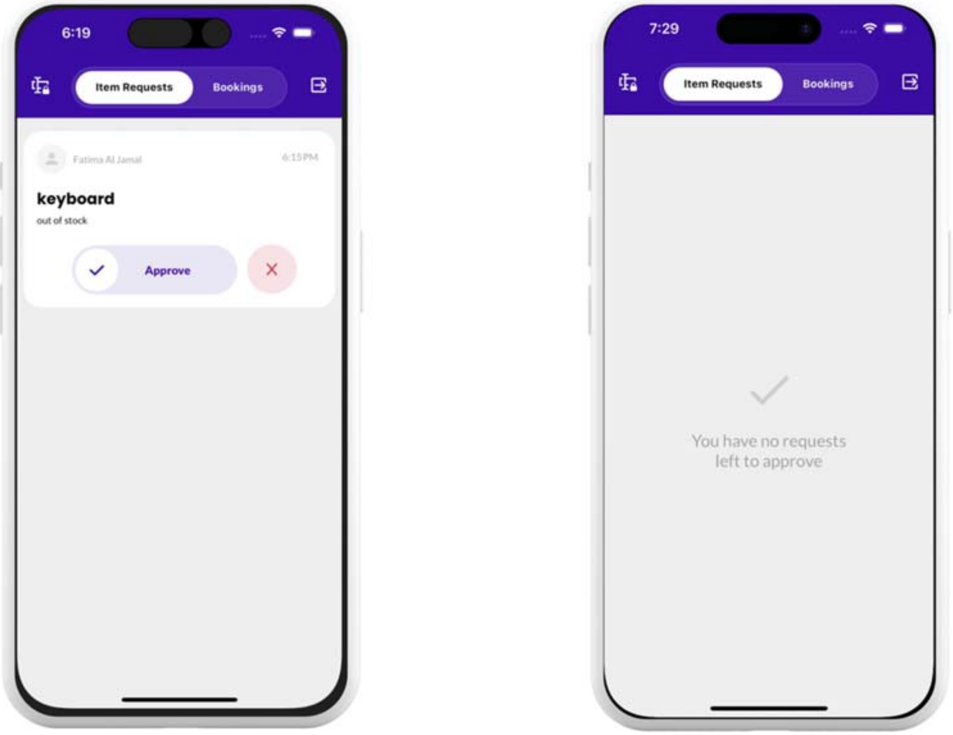
need a mouse that is more ergonomic

This request will be rejected. Continue?

Reason



You have no requests left to approve



Fatima Al Jamal 6:15PM

keyboard

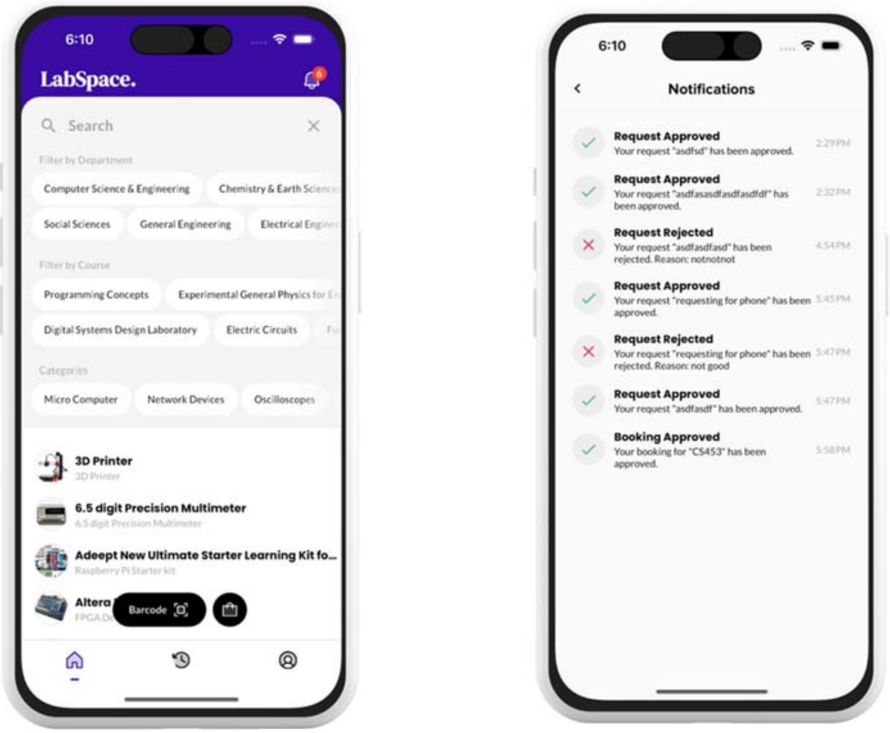
out of stock

You have no requests left to approve

Figure 25. Request new equipment process

Notifications:

User can navigate to the first tab, and there they can click the “notification” button on the top right to check where they have borrowed, approved or rejected.



✓ **Request Approved** Your request "asdfsd" has been approved. 2:29 PM

✓ **Request Approved** Your request "asdfasasdfasdfasdfdf" has been approved. 2:32 PM

✕ **Request Rejected** Your request "asdfasdfasd" has been rejected. Reason: notnotnot 4:54 PM

✓ **Request Approved** Your request "requesting for phone" has been approved. 5:45 PM

✕ **Request Rejected** Your request "requesting for phone" has been rejected. Reason: not good 5:47 PM

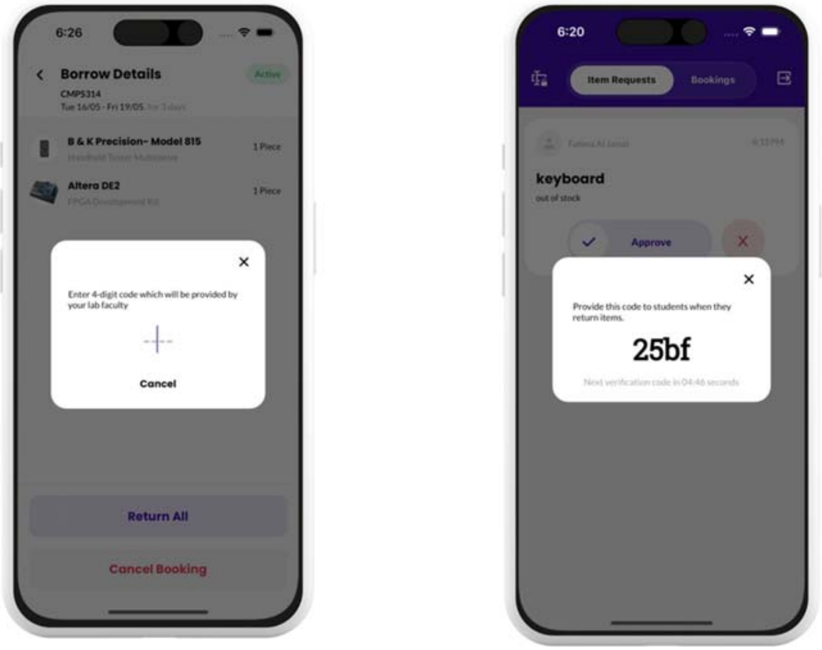
✓ **Request Approved** Your request "asdfasdf" has been approved. 5:47 PM

✓ **Booking Approved** Your booking for "CS453" has been approved. 5:58 PM

Figure 26. Notifications

Return equipment:

The user can click the icon ‘’Return All” in the bottom and get the code from faculty.



Enter 4-digit code which will be provided by your lab faculty

Provide this code to students when they return items.

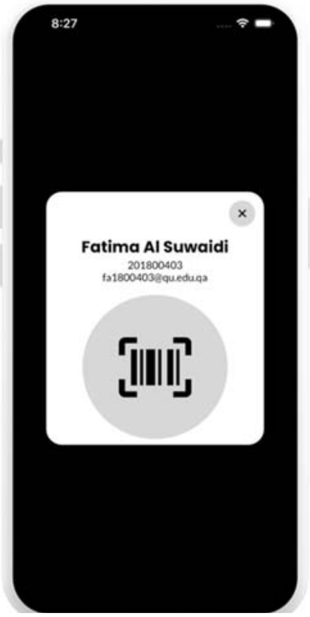
25bf

Next verification code in 04:46 seconds

Figure 27. Return equipment

Scan Student id:

The admin can scan the user id from the scan tab in the main page.

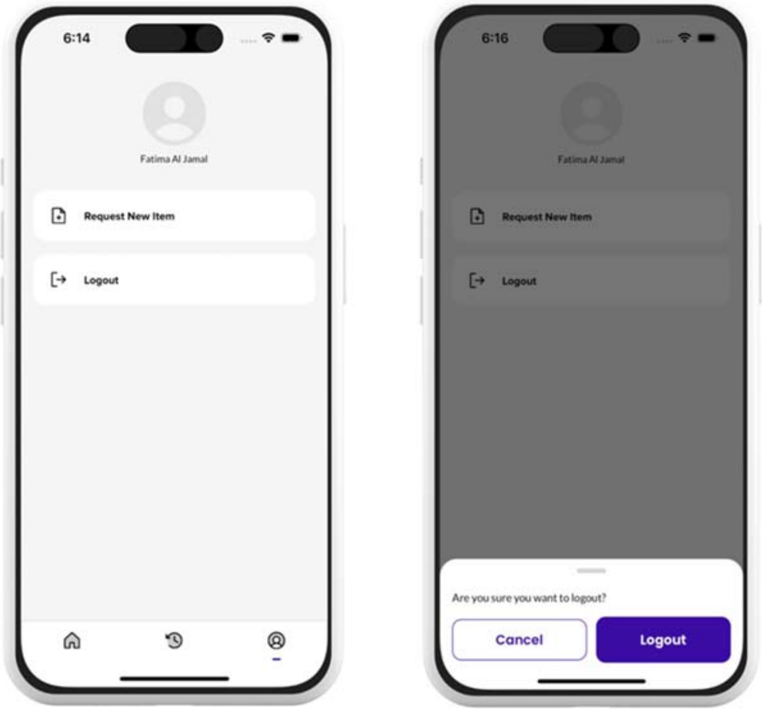


Fatima Al Suwaidi
201800403
fa1800403@qu.edu.qa

Figure 28. Scanner

Log out:

User can navigate to the last tab, and there they can click the “logout” button to be logged out.



Are you sure you want to logout?

Figure 29. Sign-out

5.6. Design patterns

MVC pattern:

The Model-View-Controller (MVC) pattern is a popular architectural design pattern which we used to simplify development in our system. The Model contains the data structure of the application, the View is responsible for displaying the data to the user while taking care of fonts, colors, spacing, etc., and the Controller contains all the business logic. This separation of concerns increased modularity, maintainability, and reusability of our code significantly.

We explored other patterns that were popular in the Flutter community such as Bloc pattern (Business Logic Component), Provider pattern(https://www.patterns.dev/posts/provider-pattern), Redux(https://medium.com/@rajeswari3699/flutter-redux-state-management-with-redux-in-flutter-7a6a13515f69) and Singleton. While we have used Singleton pattern in some parts of our code to maintain global references to variables that were commonly used, such as variables related to networking and authentication, we found that other patterns required higher learning curve and affected our productivity and decided against using them.

We used the Getx(https://pub.dev/packages/get) state management library in Flutter to achieve MVC pattern as it is not a native feature of Flutter framework. Getx allows navigation between screens without using Context object, which was a limitation of Flutter and forced logic related to screen navigation to be placed in View/User Interface part of the codebase.

Below diagram shows an example of how BagScreen was structure into MVC. We have a class definition for Bag model, which is instantiated in BagController and used in BagScreen to display the ListView for Bag. This pattern is used across other screens too, such as ApprovalScreen, BookingDetailsScreen and NotificationsScreen in a similar manner.

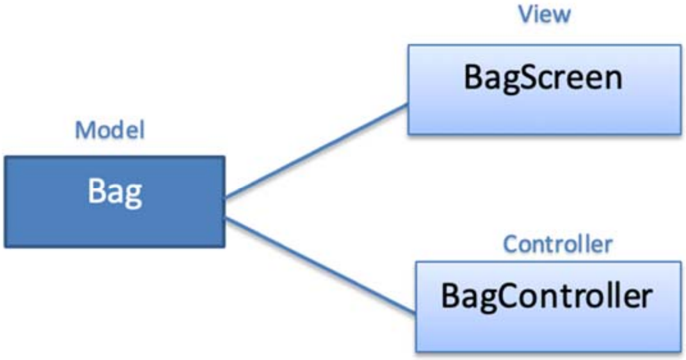


Figure 30. MVC Pattern

6. Implementation

6.1. Hardware/software used.

The suggested solution will only be developed using desktop and mobile hardware. However, we will be utilizing a variety of software programs when creating the suggested system. Their names, descriptions, and justifications of use are provided in the table below.

Table 9.Hardware/software to be used.

HW/SW details	Justifications
1. ClickUp	We will use ClickUp to share the project and be able to work as a team.
2. Flutter	We will use flutter to make the software application and develop it.
3. Visual Studio Code	It will be used to create the application and develop it.

4. Visual Paradigm	We will use Visual Paradigm in diagramming our use cases.
5. GitHub GitHub	We will use GitHub to share and save code easily with the team members
6. Google API Google APIs	We will be using Google Map in order to easily find the live location of the properties in the system
7. Google Forms	We will use Google forms to create our survey.

8. Homebrew Homebrew The missing package manager for macOS (or Linux)	We will use Homebrew to install and update applications on a Mac easily.
9. Directus 	We will use this to store all the booking information and requests. Also all the user management will be done in this platform such as management of roles and Permissions or registration of new users, in addition to the mobile app.
10. Xcode 	We will use Xcode for compiling applications for iPhone.

11. Android studio	We will use Android studio for compiling applications for android smart phones.

6.2. Challenging issues and solutions

Our project team met, and we individually decided to test all aspects of the system and then compile our findings in order to find solutions and mitigations. One challenging issue encountered during the implementation of LabSpace was getting used to using the new system. This was partly due to its user interface not being as intuitive as initially thought. Several of us found it initially challenging to navigate the system, locate equipment in order to use the borrowing process. To address this, the overall user-friendliness of the system was improved by incorporating team feedback and suggestions. Specifically, the system was updated with a more streamlined interface that allows users to find equipment more quickly and easily by cancelling a few unnecessary steps in the application. Moreover, we attempted using a phone application barcode scanning system to streamline the entire borrowing process, making it quicker and more convenient for users. However, some users found it challenging to use the Phone barcode scanners, especially those who were not familiar with the technology. The system required the phones to be updated to the latest software in order to use it effectively without crashing. Also, to overcome this, additional training sessions will be provided in the future for all staff that are responsible for this task, and the system will be updated with instructional videos on how to use the barcode scanners. We hope to eventually purchase and incorporate a physical barcode scanning device for use by university staff for more efficiency and speed. The barcode system also allowed for efficient inventory tracking, ensuring that equipment is never lost or misplaced. Overall, through these efforts, the system has become more user-friendly and will continue to improve once the actual barcode device is implemented.

Furthermore, the implementation of the LabSpace system has reinforced the importance of incorporating user feedback and suggestions during the development process. In the future, we plan to do a "mock testing phase", by gathering a few more of our classmates and some university staff to attempt using the system and providing their feedback. By seeking input from users, potential issues or hindrances can be identified and addressed, resulting in a more effective and user-friendly system. Additionally, the project highlighted the significance of thorough testing and addressing any technical issues that arise promptly. With our team members initial brainstorming, we managed to improve on at least 5 problems that we did not see in earlier stages. Another valuable lesson learned is the importance of providing adequate training and support to users in the future, particularly during the initial implementation phase. By providing continuous training, users can quickly become proficient in using the system, leading to greater adoption and success. Overall, the LabSpace project has taught us the importance of user-centered design, thorough testing, and support in the successful implementation of technological solutions.

7. Testing and evaluation

7.1. Functional testing

Table 10. Functional testing

Test Case No.	Description	Related Use Case	Steps	Expected Result	Actual Result
1	Be able to register a user of “Student” role by entering all the necessary details.	Sign up	1. Tap on “Don’t have an account?” 2. Enter preferred name, email, and password 3. Tap on Sign-Up button.	Upon entering the details in correct format, the sign-up button gets enabled. Upon tapping the button, a student account is created and message is shown.	Matches expected result. (Figure 31)
2	Be able to sign in as a user of any role from the same screen and be redirected to the appropriate screen.	Sign in	1. Enter email 2. Enter password 3. Tap on the Sign-In button	Upon entering correct credentials, Lab Faculty and Head of Department to be taken to approval screens, Students to be taken to equipment booking screens.	Matches expected result. (Figure 32)
3	Be able to sign out of the application	Sign Out	1. Navigate to the last tab. 2. Click on “Log Out” button	Upon tapping “Log Out” button the user is signed out and unable to perform any action in the app, other than signing in again.	Matches expected result. (Figure 33)
4	Be able to reset password from the portal	Reset Password	1. Open the web portal. 2. Click on “Forgot Password.” 3. Enter email. 4. Click on the link in the received email 5. Enter new password	Upon clicking confirm button the password is reset, and the user is able to sign into the app with the new password.	Matches expected result. (Figure 34)

			6. Click confirm		
5	Head of Department or Lab Faculty is able to Approve or Reject bookings and equipment requests.	Process Request	1. Sign-in to the mobile app 2. Pick a request or booking. 3. Swipe to Approve and click “x” button to Reject.	Upon Approving or Rejecting the request/booking disappears from the list and notification is sent to the user linked with the request/booking.	Matches expected result. (Figure 35)
6	When requesting for a new item, user can add a comment to explain the request.	Add Comment	1. Sign-in to the mobile app 2. Navigate to the last tab 3. Tap on “Request New Item” 4. Enter comment	Upon creating the request, the comment is visible to the Head of Department or Faculty.	Matches expected result. (Figure 36)
7	A student can view all bookings they made from certain place in the app	View all bookings	1. Sign-in to the mobile app 2. Navigate to the second tab 3- View bookings and tap on any to view more details	User can see all bookings and their respective statuses from the same screen. They can also view more details by tapping on any booking.	Matches expected result. (Figure 37)
8	User can view request status	View Request Status	<u>Students</u> 1. Sign-in to the mobile app 2. Navigate to Notifications screen by clicking on Bell icon <u>Faculty</u> 1. Sign-in to the portal 2. Navigate to “Requests”	Users, both students and faculties can view the status of the requests made by them	Matches expected result. (Figure 38)
9	A faculty can scan the student card to retrieve all information about the	Scan Student Card	1. Sign-in to the mobile app 2. Click on “Scan Barcode” button	Upon scanning the card, student details stored in the system are displayed	Matches expected result. (Figure 39)

	student stored in LabSpace database		3. Scan barcode on a student ID card		
10	A student can view all details of a booking including start date, end date, duration, all the items and quantities	View Booking Details	1. Sign-in to the mobile app 2. Navigate to the second tab. 3. View bookings and tap on any to view more details	User can see all booking details on which they tapped	Matches expected result. (Figure 40)
11	A user of any role can view equipment details	View Item Details	<u>Student</u> 1. Login to the mobile app 2. Click on any equipment <u>Faculty</u> 1. Login to the portal 2. Click on "Equipments" 3. Click on any equipment	User can view all details of the equipment such as description, and whether it is available or not	Matches expected result. (Figure 41)
12	A student or faculty can request a new item	Request New Item	<u>Students</u> 1. Sign-in to the mobile app 2. Navigate to the last tab 3. Click on "Request New Item." 4. Enter details 5. Click on submit button <u>Faculty</u> 1. Sign-in to the portal 2. Navigate to "Requests." 3. Click on "+" button 4. Enter details	Upon clicking submit button the request is created, and confirmation message appears for the user	Matches expected result. (Figure 42)

			5. Click on submit button		
13	A student can view history of all bookings they made	View Booking History	1. Sign-in to the mobile app 2. Navigate to the second tab. 3. View bookings and tap on any to view more details	User can see all bookings and their respective statuses from the same screen. They can also view more details by tapping on any booking.	Matches expected result. (Figure 43)
14	User is asked to accept an agreement. If not agreed, the user is not allowed to borrow items	Accept Agreement	1. Login to the mobile app 2. Complete all the steps of Book an Item use case, until the last step. 3. Tick the agreement checkbox.	If the user does not agree to the agreement borrowing button is greyed out. Otherwise, the user can continue borrowing	Matches expected result. (Figure 44)
15	User can book an item by selecting the time period and quantity of the item	Book an Item	1. Login to the mobile app 2. Tap on desired equipment 3. Tap “Add to Bag.” 4. Then go to the Bag and adjust quantities of the equipments 5. Tap on “Borrow All.” 6. Select the desired time period, i.e., Start Date and End Date, or tap on one of the 1-click suggestions. Then, tap Next. 7. The app will check for availability of the	The booking is created, and a confirmation message is displayed. Additionally, the booking can be viewed in the booking history page.	Matches expected result. (Figure 45)

			equipments during the time period. Click on “Continue” once the scan is completed. 8. Enter the purpose of borrowing and tick the “I agree” checkbox		
16	User can search equipments from the search bar	Search Item	Student 1. Login to the mobile app 2. Enter equipment name in the search bar Faculty 1. Login to the portal 2. Click on “Equipments.” 3. Enter equipment name in the search bar	The equipment appears in the search results if it exists in the database.	Matches expected result. (Figure 46)
17	User can return item after they are done using the item.	Return an Item	1. Login to the mobile app 2. Go to the Booking History tab 3. Choose the booking in which the item is 4. Click on “Return All” button 5. Enter the confirmation code from the Lab Faculty	Upon entering correct confirmation code, the booking status changes to “Returned” and a confirmation message is displayed to the user	Matches expected result. (Figure 47)

18	User can cancel a booking if it is not already approved	Cancel Booking	1. Login to the mobile app 2. Go to the Booking History tab 3. Choose the booking in which the item is 4. Click on "Cancel Booking" button 5. Enter the confirmation code from the Lab Faculty	Confirmation message is displayed and the status of the booking changes to "Cancelled".	Matches expected result. (Figure 48)
19	User can create a new lab equipment in the system	Create New equipment	Faculty (This action is allowed for Faculties only) 1. Login to the portal 2. Go to the "Equipments" tab 3. Click on "+" button. 4. Enter all equipment details in the form. Make sure to fill all required fields. 5. Upload equipment picture 6. Click on Save button	Confirmation message is displayed to the user and user can view the item in the equipments list from both the App and Portal	Matches expected result. (Figure 49)

20	User can generate reports to gain further Business Intelligence	Generate reports	Faculty (This action is allowed for Faculties only) 1. Login to the portal 2. Go to the “Bookings” tab, or any other tab of which user would like to view the Report of. 3. On the right-side bar, click on “Export Items.” 4. Choose desired format:. csv,. json 5. Choose the fields to be included in the report 6. After selecting the desired settings click on the “Download” button in the top-right corner of the screen.	A report will be generated in the chosen format and then downloaded in the user’s browser’s Downloads folder. If it is .csv file, it can be opened in Excel.	Matches expected result. (Figure 50)
21	User can archive an item that is no longer in use, so that it is not displayed anymore in the mobile app	Archive Item	Faculty (This action is allowed for Faculties only) 1. Login to the portal 2. Go to the “Equipments” tab 3. Select an equipment 4. Update the status of the equipment to “Archived” from “Active.”	A confirmation message will be displayed to the user. In addition, the equipment will not be visible anymore in the mobile application	Matches expected result. (Figure 51)

			5. Click on Save button		
22	User can edit the details of an existing equipment or upload a new picture	Edit equipment	Faculty (This action is allowed for Faculties only) 1. Login to the portal 2. Go to the “Equipments” tab 3. Select an equipment 4. Update details of the equipment 5. Click on Save button	User will be able to see the updated details of the equipment reflected in the list, alongside a confirmation message	Matches expected result. (Figure 52)
23	User can view all requests in the system	View all request	Faculty (Viewing all requests is allowed for Faculties only) 1. Login to the portal 2. Go to “Requests” tab	User is able to view all requests	Matches expected result. (Figure 53)

24	User can update details of his/her own profile	Update profile	1. Login to the mobile application 2. Go to the last tab 3. Click on user’s avatar 4. Update details as desired 5. Click on the tick icon to save	User profile is updated, and user can immediately see the new details reflected below the user avatar	Matches expected result. (Figure 54)
25	User can view his/her own profile	View profile	1. Login to the mobile application 2. Go to the last tab 3. Click on user’s avatar	User will be able to see all profile details	Matches expected result. (Figure 55)

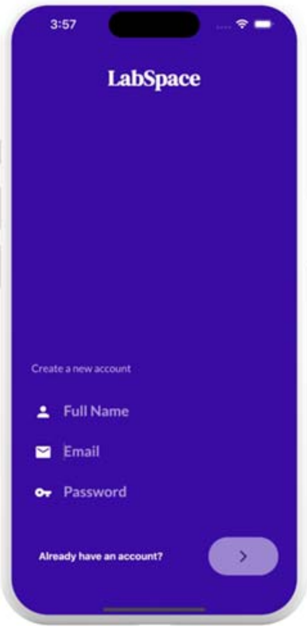


Figure 32. Test case 1, Sign-up

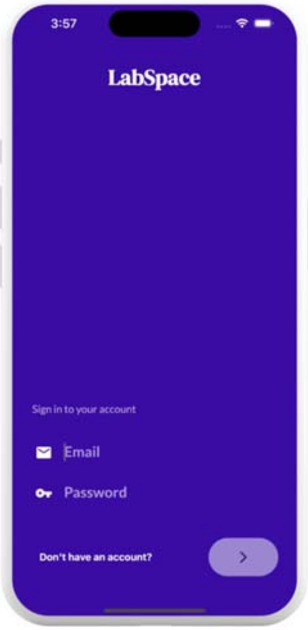


Figure 31. Test case 2, Sign-in

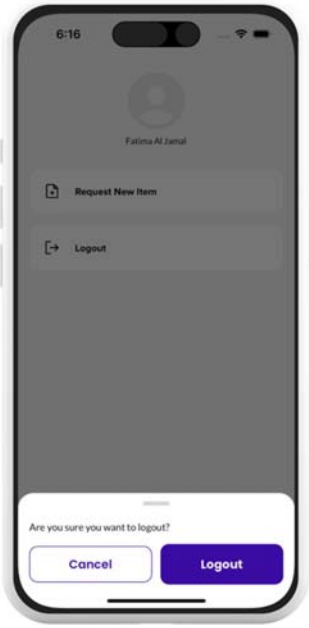
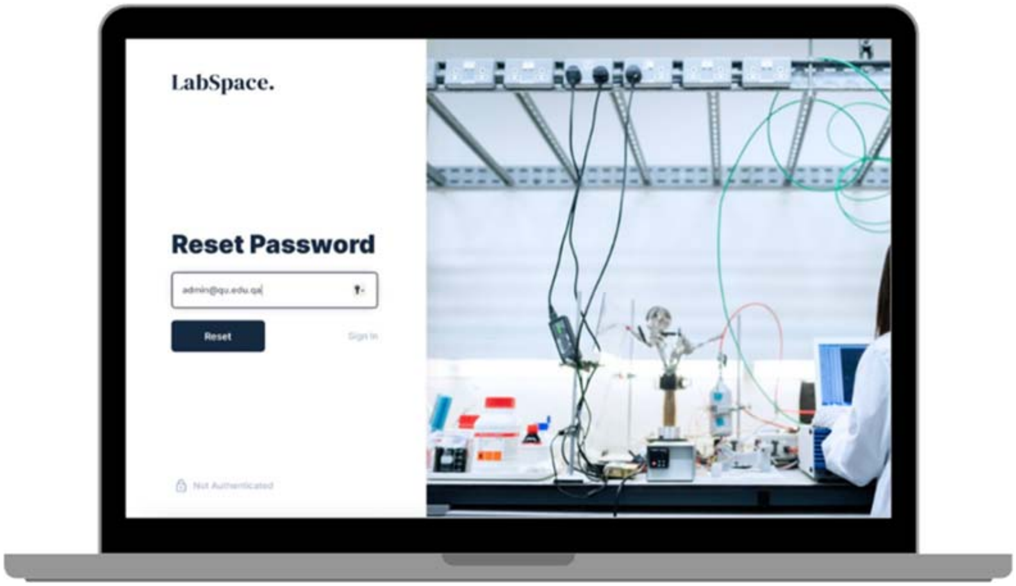


Figure 33. Test case 3, Sign-out



admin@qu.edu.qa

Figure 34. Test case 4, Reset password

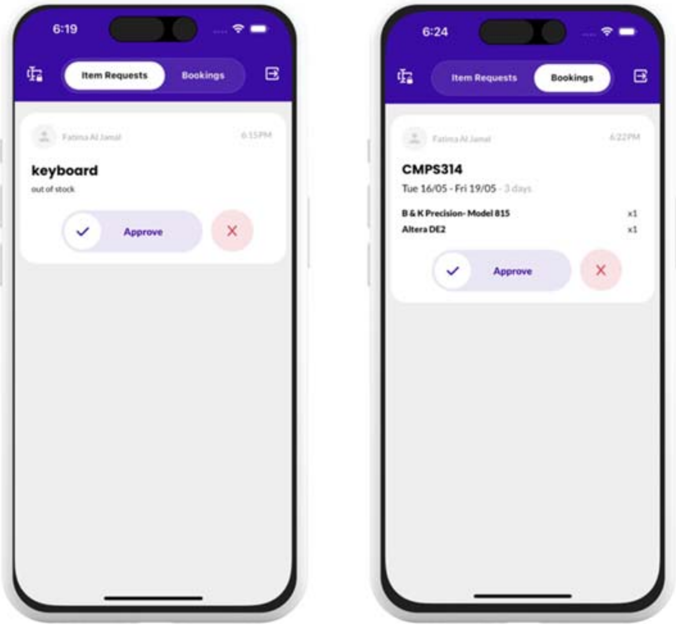


Figure 35. Test case 5, Process request

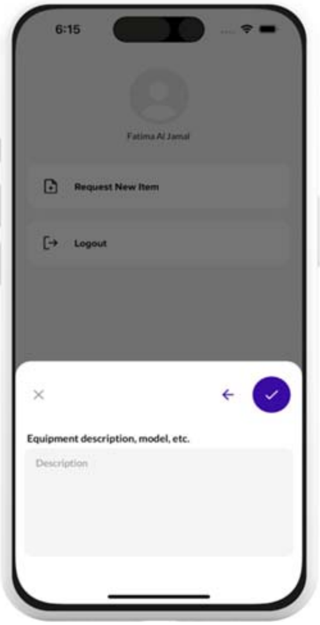
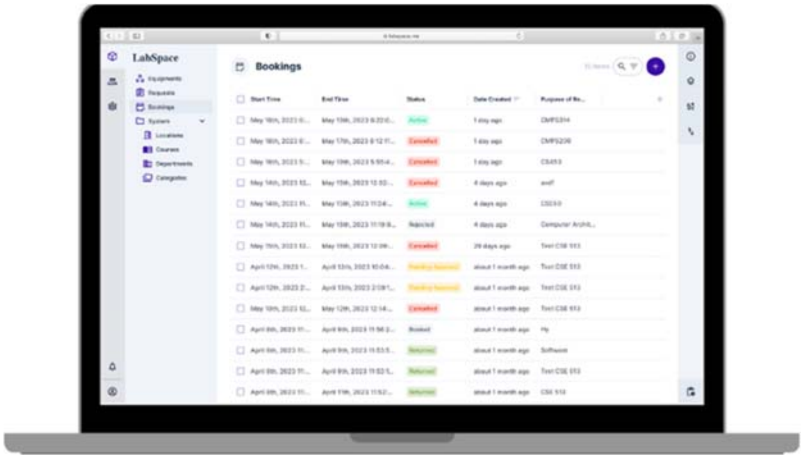
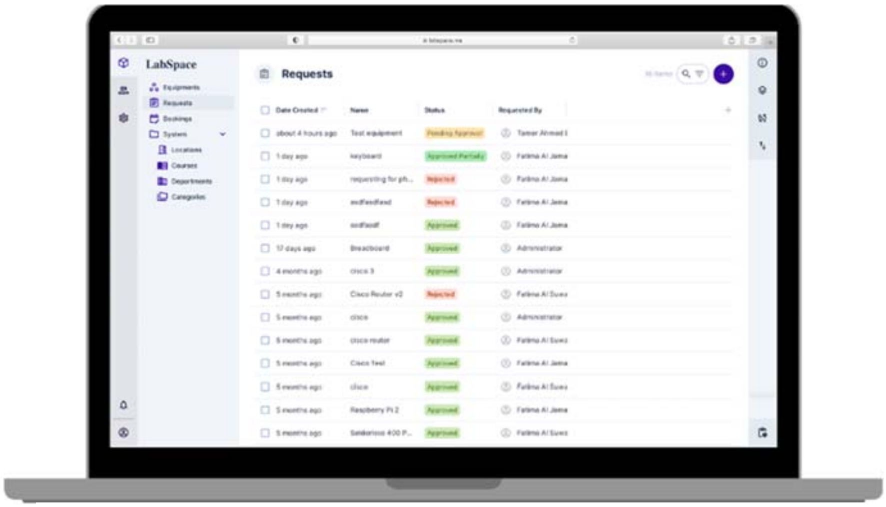


Figure 36. Test case 6, Add comment



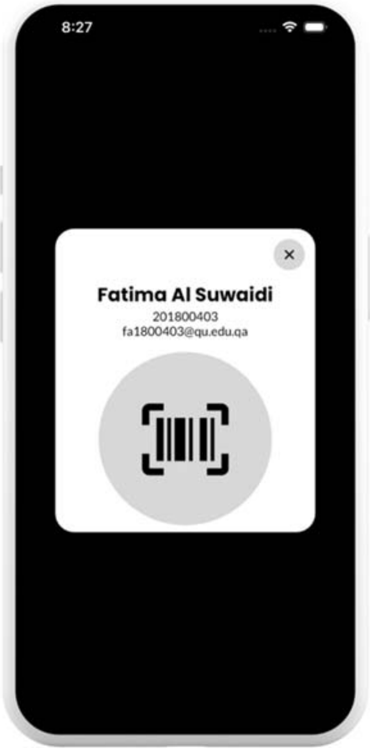
Start Time	End Time	Status	Date Created	Purpose of Bo...
May 16th, 2023 6:...	May 19th, 2023 6:22:0...	Active	1 day ago	CMPS314
May 16th, 2023 6:...	May 17th, 2023 6:12 11...	Cancelled	1 day ago	CMPS206
May 16th, 2023 5:...	May 19th, 2023 5:55:4...	Cancelled	1 day ago	CS453
May 14th, 2023 12:...	May 15th, 2023 12:32:...	Cancelled	4 days ago	asdf
May 14th, 2023 11:...	May 13th, 2023 11:24:...	Active	4 days ago	CS033
May 14th, 2023 11:...	May 15th, 2023 11:19:8...	Rejected	4 days ago	Computer Archit...
May 11th, 2023 12:...	May 19th, 2023 12:09:...	Cancelled	29 days ago	Test CSE 513
April 12th, 2023 1...	April 13th, 2023 10:04:...	Pending Approval	about 1 month ago	Test CSE 513
April 12th, 2023 2...	April 13th, 2023 2:08:1...	Pending Approval	about 1 month ago	Test CSE 513
May 10th, 2023 12:...	May 12th, 2023 12:14:...	Cancelled	about 1 month ago	Test CSE 513
April 8th, 2023 11:...	April 9th, 2023 11:56:2...	Booked	about 1 month ago	Hy
April 8th, 2023 11:...	April 9th, 2023 11:53:5...	Returned	about 1 month ago	Software
April 8th, 2023 11:...	April 9th, 2023 11:53:1...	Returned	about 1 month ago	Test CSE 513
April 8th, 2023 11:...	April 11th, 2023 11:52:...	Returned	about 1 month ago	CSE 513

Figure 37. Test case 7, View all bookings



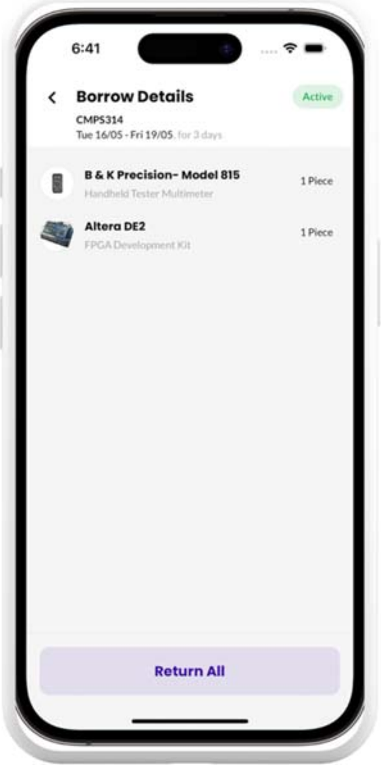
Date Created	Name	Status	Requested By
about 4 hours ago	Test equipment	Pending Approval	Tamer Ahmed I
1 day ago	keyboard	Approved Partially	Fatima Al Jama
1 day ago	requesting for ph...	Rejected	Fatima Al Jama
1 day ago	asdfasdfasd	Rejected	Fatima Al Jama
1 day ago	asdfasdf	Approved	Fatima Al Jama
17 days ago	Breadboard	Approved	Administrator
4 months ago	cisco 3	Approved	Administrator
5 months ago	Cisco Router v2	Rejected	Fatima Al Suwa
5 months ago	cisco	Approved	Administrator
5 months ago	cisco router	Approved	Fatima Al Suwa
5 months ago	Cisco Test	Approved	Fatima Al Jama
5 months ago	clisco	Approved	Fatima Al Suwa
5 months ago	Raspberry Pi 2	Approved	Fatima Al Jama
5 months ago	Seldorious 400 P...	Approved	Fatima Al Suwa

Figure 40. Test case 8, View request status



Fatima Al Suwaidi
201800403
fa1800403@qu.edu.qa

Figure 38. Test case 9, Scan student card

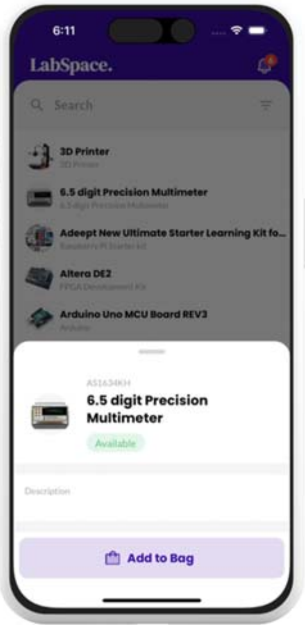


Borrow Details — Active

CMPS314
Tue 16/05 - Fri 19/05, for 3 days

B & K Precision- Model 815 Handheld Tester Multimeter	1 Piece
Altera DE2 FPGA Development Kit	1 Piece

Figure 39. Test case 10, View booking details



3D Printer
3D Printer

6.5 digit Precision Multimeter
6.5 digit Precision Multimeter

Adeept New Ultimate Starter Learning Kit fo...
Raspberry Pi Starter kit

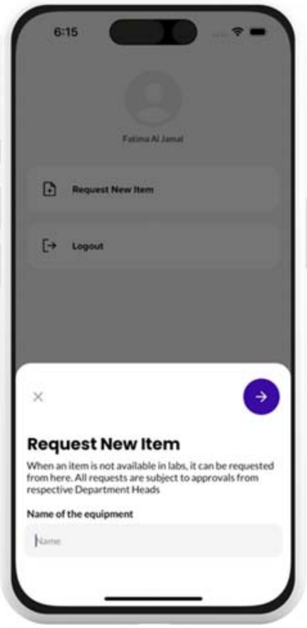
Altera DE2
FPGA Development Kit

Arduino Uno MCU Board REV3
Arduino

AS1634KH
6.5 digit Precision Multimeter
Available

Description

Figure 42. Test case 11, View item details



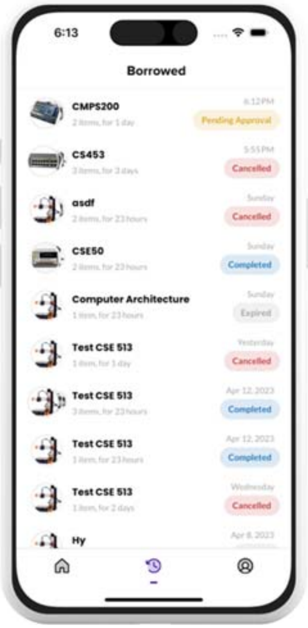
Request New Item

When an item is not available in labs, it can be requested from here. All requests are subject to approvals from respective Department Heads

Name of the equipment

Name

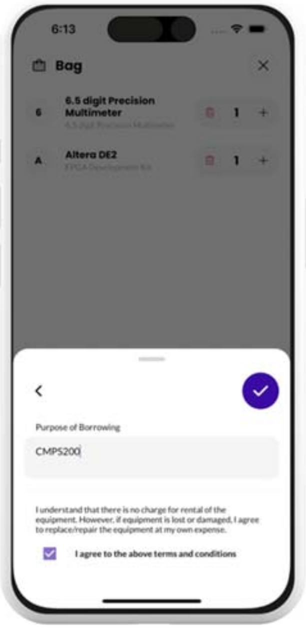
Figure 43. Test case 12, Request new item



Borrowed

CMPS200 — 2 items, for 1 day — 6:12 PM — Pending Approval
CS453 — 3 items, for 3 days — 5:55 PM — Cancelled
asdf — 2 items, for 23 hours — Sunday — Cancelled
CSE50 — 2 items, for 23 hours — Sunday — Completed
Computer Architecture — 1 item, for 23 hours — Sunday — Expired
Test CSE 513 — 1 item, for 1 day — Yesterday — Cancelled
Test CSE 513 — 3 items, for 23 hours — Apr 12, 2023 — Completed
Test CSE 513 — 1 item, for 23 hours — Apr 12, 2023 — Completed
Test CSE 513 — 1 item, for 2 days — Wednesday — Cancelled
Hy — Apr 8, 2023

Figure 41. Test case 13, View booking history



6 — 6.5 digit Precision Multimeter — 6.5 digit Precision Multimeter — 1
A — Altera DE2 — FPGA Development Kit — 1

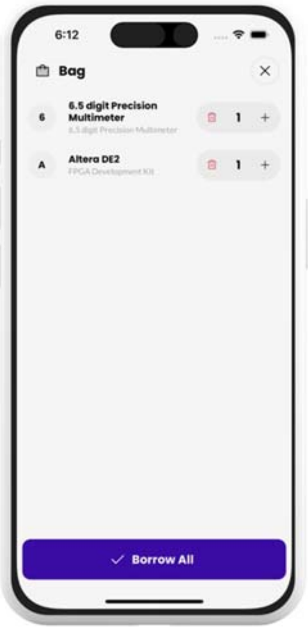
Purpose of Borrowing

CMPS200

I understand that there is no charge for rental of the equipment. However, if equipment is lost or damaged, I agree to replace/repair the equipment at my own expense.

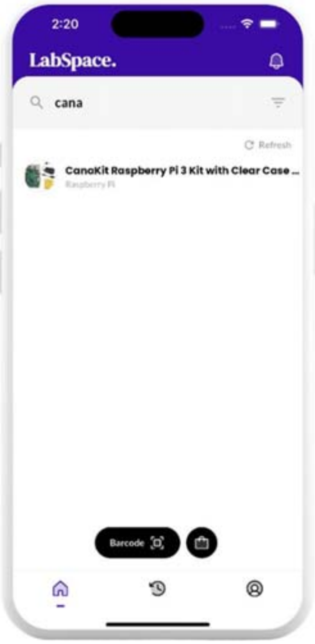
☑ I agree to the above terms and conditions

Figure 44. Test case 14, Accept agreement



6 — 6.5 digit Precision Multimeter — 6.5 digit Precision Multimeter — 1
A — Altera DE2 — FPGA Development Kit — 1

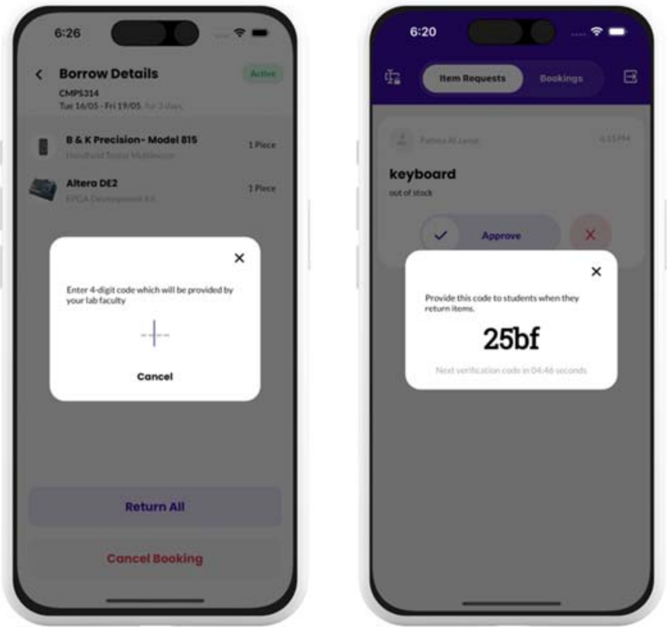
Figure 46. Test case 15, Book an item



cana

CanaKit Raspberry Pi 3 Kit with Clear Case ...
Raspberry Pi

Figure 45. Test case 16, Search item



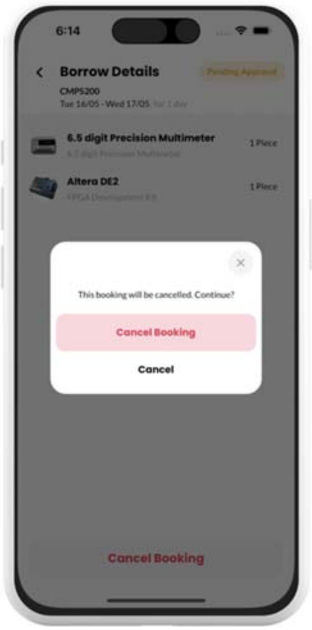
Enter 4-digit code which will be provided by your lab faculty

Provide this code to students when they return items.

25bf

Next verification code in 04:46 seconds

Figure47 . Test case 17, Return an item



This booking will be cancelled. Continue?

Figure48 . Test case 18, Cancel booking

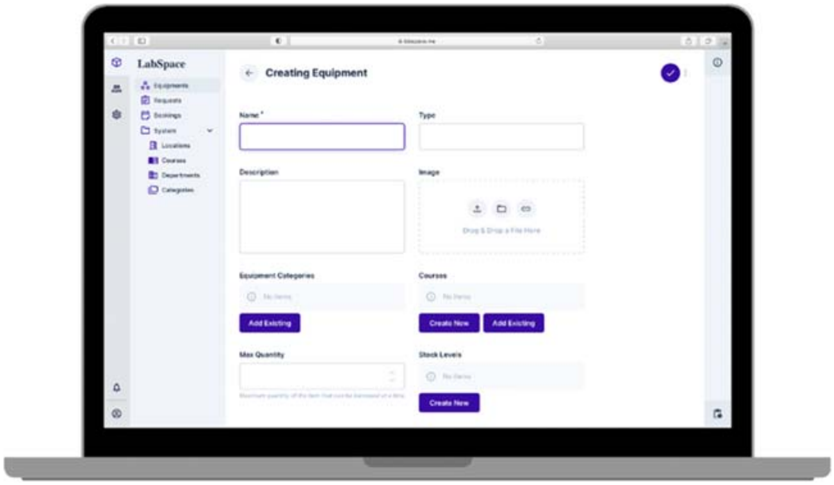
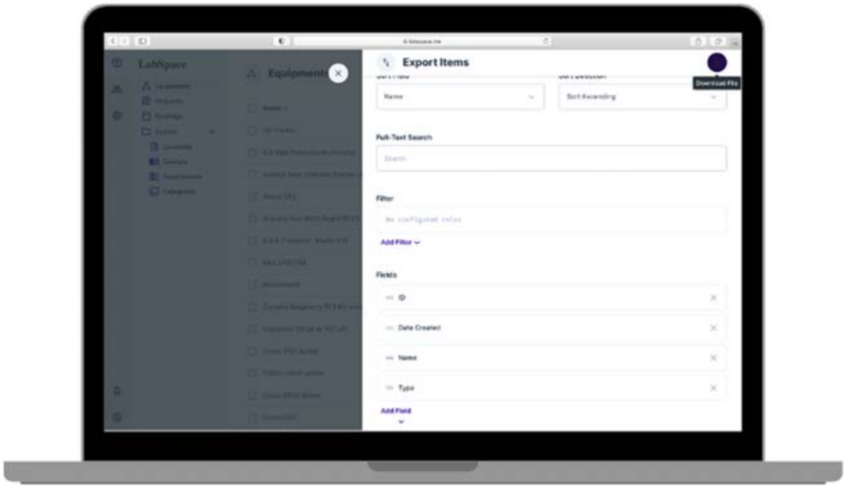


Figure 49. Test case 19, Create new equipment



Export Items

Sort Field: Name

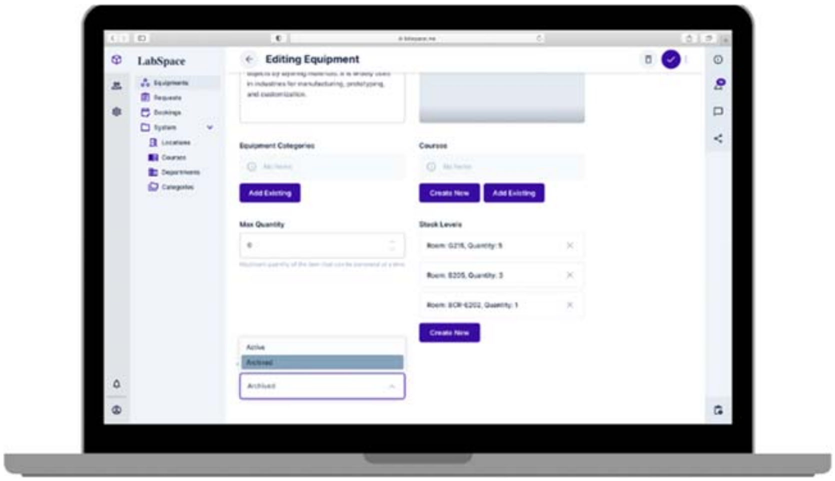
Sort Direction: Sort Ascending

Full-Text Search: Search

Filter: No configured rules

Fields: ID, Date Created, Name, Type

Figure 51. Test case 20, Generate reports



Editing Equipment

objects by layering materials. It is widely used in industries for manufacturing, prototyping, and customization.

Equipment Categories: No items

Courses: No items

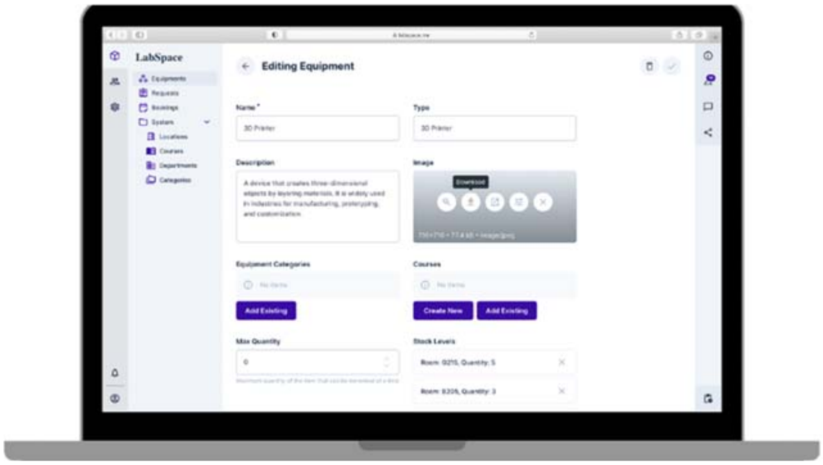
Max Quantity: 0

Maximum quantity of the item that can be borrowed at a time

Stock Levels:
Room: G215, Quantity: 5
Room: E205, Quantity: 3
Room: ECR-E202, Quantity: 1

Archived

Figure 52. Test case 21, Archive item



Editing Equipment

Name *: 3D Printer

Type: 3D Printer

Description: A device that creates three-dimensional objects by layering materials. It is widely used in industries for manufacturing, prototyping, and customization.

Image

Equipment Categories: No items

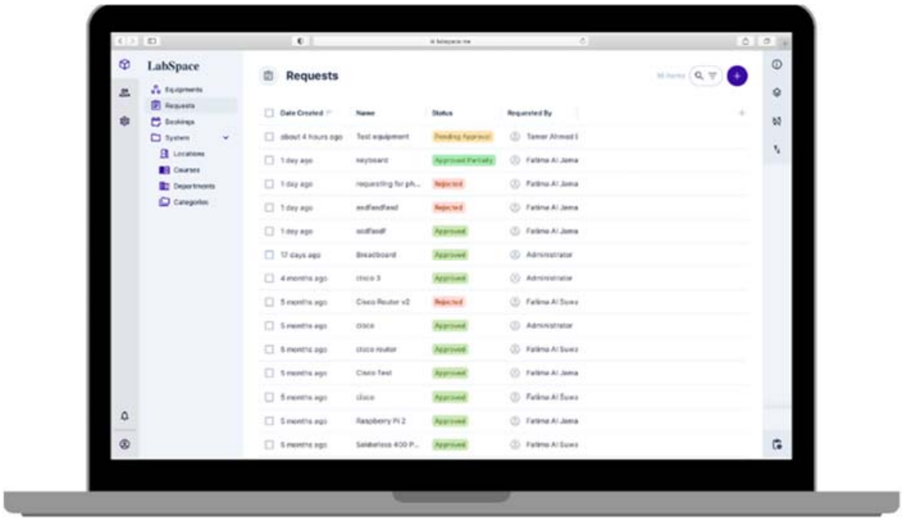
Courses: No items

Max Quantity: 0

Maximum quantity of the item that can be borrowed at a time

Stock Levels:
Room: G215, Quantity: 5
Room: E205, Quantity: 3

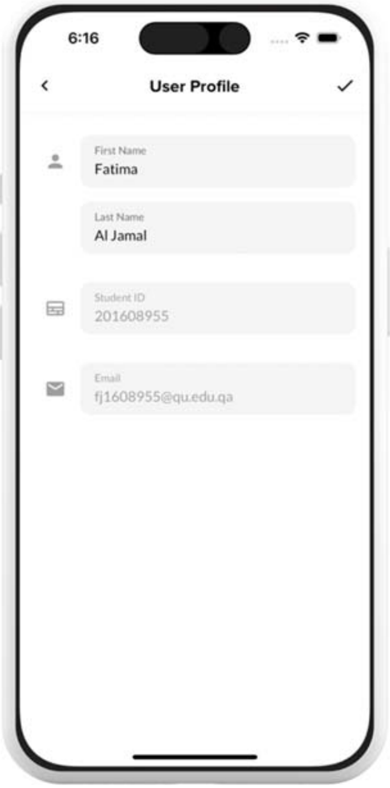
Figure 50. Test case 22, Edit equipment



Requests

□	Date Created	Name	Status	Requested By
□	about 4 hours ago	Test equipment	Pending Approval	Tamer Ahmed I
□	1 day ago	keyboard	Approved Partially	Fatima Al Jama
□	1 day ago	requesting for ph...	Rejected	Fatima Al Jama
□	1 day ago	asdfasdfasd	Rejected	Fatima Al Jama
□	1 day ago	asdfasdf	Approved	Fatima Al Jama
□	17 days ago	Breadboard	Approved	Administrator
□	4 months ago	cisco 3	Approved	Administrator
□	5 months ago	Cisco Router v2	Rejected	Fatima Al Suwa
□	5 months ago	cisco	Approved	Administrator
□	5 months ago	cisco router	Approved	Fatima Al Suwa
□	5 months ago	Cisco Test	Approved	Fatima Al Jama
□	5 months ago	cisco	Approved	Fatima Al Suwa
□	5 months ago	Raspberry Pi 2	Approved	Fatima Al Jama
□	5 months ago	Solderless 400 P...	Approved	Fatima Al Suwa

Figure 53. Test case 23, View all request



User Profile

First Name
Fatima

Last Name
Al Jamal

Student ID
201608955

Email
fj1608955@qu.edu.qa

Figure 54. Test Case 24, Update profile

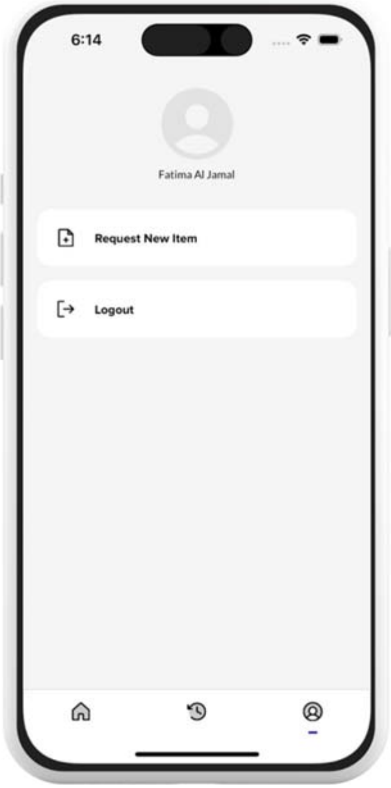


Figure 55. Test case 25, View profile

7.2. Non-functional testing

In this section, we present the tests conducted to evaluate the quality of the system and verify that it satisfies the non-functional requirements of the project. We have performed various tests to assess the system's usability, performance, modifiability, security, and reliability. The following sub-sections describe each non-functional requirement in detail and present the corresponding test results.

1. Usability

The usability of the system was evaluated to ensure it is easy to use with minimal steps required. The onboarding process for first-time users was assessed, aiming for completion within 2 minutes. Additionally, user feedback was gathered to identify areas for UI/UX improvements.

Testing Method:

1. Conducted a user-testing session with a minimum of 20 end-users.
2. Measured the time taken by each user to complete primary scenarios of the system.
3. Collected feedback from users using the below questionnaire:
    a. How was your overall experience while using the app?
        i. Positive
        ii. Neutral
        iii. Negative
    b. Do you have any suggestions for improvements?

User	Onboarding Time (minutes)	Overall Feedback
1	1.5	Positive
2	2.1	Neutral
3	1.8	Positive
4	2.3	Negative

5	1.6	Positive
6	1.9	Neutral
7	2.2	Positive
8	2.0	Positive
9	1.7	Positive
10	2.4	Negative
11	1.8	Positive
12	2.1	Neutral
13	1.9	Positive
14	2.3	Negative
15	1.6	Positive
16	1.7	Positive
17	2.0	Positive
18	1.5	Positive
19	2.2	Positive
20	1.8	Positive

The test results indicate that the system meets the usability requirements, as the average onboarding time for first-time users was within the specified 2-minute limit. The collected feedback was used to further improve the UI/UX aspects of the system.

2. Performance Testing

Performance testing was conducted to ensure the system is responsive and meets the loading time requirements. Multiple tests were performed for each use-case's scenarios under various circumstances, such as poor internet connection or slow hardware devices. The maximum threshold set for loading times was 10 seconds.

Scenario No.	Scenario	Loading Time (seconds)	Result
Scenario 1	Normal network	6.2	Passed
Scenario 2	Slow network	12.5	Failed

Scenario 3	Normal hardware	4.8	Passed
Scenario 4	Slow hardware	9.6	Passed

Based on the test results, the system meets the performance requirements in most scenarios. However, in the "Slow network" scenario of Scenario 1, the loading time exceeds the maximum threshold. Further optimizations will be implemented to improve performance under such circumstances.

3. Modifiability Testing

Modifiability testing was conducted to assess the system's ability to accommodate evolving requirements with minimal cost. A change in one of the use-case requirements was simulated, and the effort required for the modification was measured.

No.	**Modification**	**Effort (person-hours)**	**Result**
1	Update data validation rules	3.5	Passed
2	Add new user role	2.2	Passed
3	Modify report generation	3.1	Passed
4	Enhance search functionality	5.6	Failed
5	Modify approval workflow	6.3	Failed

The above test results indicate that the system is easily modifiable, as the modification 3 required a relatively low effort without changes in multiple places. This demonstrates that the system can adapt to changing requirements efficiently.

4. Security Testing

Security testing was conducted to ensure the protection of personal student information and college-related data. Best practices, including end-to-end encryption, were implemented to enhance security and SSL certificate used for our domain: https://labspace.me.

Test Case	Result
Login	Passed
Unauthorized access attempt	Passed
Data encryption	Passed
Secure password storage	Passed
Access control and permissions	Passed
Protection against SQL injection	Failed
Secure session management	Passed

In the above table, various test cases related to security were run. The result column indicates whether each test case passed or failed. It demonstrates the system's compliance with security requirements, such as successful login authentication, prevention of unauthorized access, data encryption, secure password storage, access control and permissions, and secure session management. We had a field that allowed injection of SQL which we have now patched.

5. Reliability Testing

Reliability testing was conducted to ensure the system's uptime and the presence of appropriate fail-over mechanisms and backup systems in case of a disaster. A system health checker was implemented to monitor the system's availability and identify any errors.

- System Uptime: The system demonstrated high uptime, with no reported downtime during the testing period. It remained accessible and responsive to user requests without any significant interruptions.
- Health Checker: The system health checker successfully pinged the backend at regular intervals and performed necessary actions. It effectively identified and

highlighted any errors or issues that occurred during the testing. Immediate notifications were generated, allowing the team to address and resolve the problems promptly.

By maintaining appropriate fail-over mechanisms and automatic backup systems, the system was able to ensure its reliability and minimize the impact of any potential disasters. The implementation of the health checker provided continuous monitoring, which will allow proactive measures to be taken to maintain system stability and availability.

Overall, the reliability testing results demonstrate that the system is robust and can withstand potential failures, ensuring uninterrupted access and performance for the end-users.

8. Conclusion

In conclusion, LabSpace system has made significant improvements to the borrowing process for engineering students and staff. During the testing phase, we found that the LabSpace system allows students to easily locate the equipment they need and request to borrow it with just a few clicks, from anywhere on campus. Administrative staff can now effectively monitor equipment usage and allocate resources efficiently. Users have reported improved satisfaction with the borrowing experience. One student noted that LabSpace saved them the hassle of going from department to department to locate equipment, while another praised the prompt approval process.

The testing process has highlighted notable achievements of the LabSpace system, such as the elimination of errors and streamlining of the borrowing process through the cancellation of manual input when borrowing equipment. The incorporation of a barcode system has resulted in more efficient and accurate tracking of equipment usage. The coding and creation of the application for booking and borrowing functions has also made the system more user-friendly, resulting in higher user satisfaction. Additionally, the alarm system coded into the system to alert administrators when inventory levels are low ensures that the system always maintains an adequate inventory level. These achievements demonstrate the potential of technology in streamlining daily activities such as borrowing equipment, and the limitless possibilities for future innovations in improving our lives.

However, there may be challenges onboarding existing students to the new system of borrowing equipment due to their familiarity with the traditional method. We also acknowledge that implementing the LabSpace system incurs initial costs such as hardware, software, and network infrastructure upgrades. During the development and implementation process, we faced some technical issues, which we were able to resolve in a timely manner.

Overall, despite these challenges, the LabSpace system has shown promise in improving the efficiency and effectiveness of equipment borrowing in an academic setting. As we prepare for rollout, we will continue to address these challenges and work towards further enhancements, ensuring that LabSpace becomes an even more robust and widely adopted system.

9. Future work

Looking to the future, the LabSpace system holds vast potential for expansion and optimization. For example, a website as well as a mobile application could be developed to allow for even more convenient access to and management of equipment. Additionally, the system could be integrated with a SMS management software to allow for real-time tracking of equipment and supplies, sending the end user SMS whenever an action has been taking (e.g.Item has been return successfully). Furthermore, the LabSpace system could be expanded to other departments or even universities, allowing for greater access to resources and collaboration between institutions. As technology advances, there will be new opportunities to enhance LabSpace and further streamline the borrowing process for engineering students and staff alike.

One of the shortcomings of the current LabSpace system is the initial training required for staff to input new equipment items into the system. Improvements could be made to simplify this process and make it more user-friendly. We can run a seminar to train staff on how to use the application features efficiently by addressing these and other potential improvements, the LabSpace system can continue to evolve and provide an even better experience to students and staff, while maximizing the utilization of resources.

10. Student reflections

Fatima ALSuwaidi

During the LabSpace project, I learned several valuable lessons that will stay with me for the rest of my career. Firstly, I learned the importance of listening to user feedback and incorporating it into the design process. Through the feedback we received, we were able to make the system more user-friendly and suitable for its intended purpose. Secondly, I developed new technical skills in problem solving, design thinking, and developing efficient solutions. These skills will be highly valuable for my future career, especially in the field of software development. Lastly, I honed my interpersonal skills such as leading a team, effective communication, and fostering collaboration between team members. Personally, I also learned the value of adapting to change and acting professionally and ethically during challenging situations.

One key shortcoming that I must avoid in future projects is not taking enough time to consider all the possible risks and challenges that might arise during the implementation stage. This can lead to unexpected setbacks and delays that could have been prevented with proper planning and preparation.

From my LabSpace project experience, one key lesson that stood out to me is the importance of team collaboration and effective communication. I believe these skills will help me in my career to work more effectively as a team member and lead to producing better quality work. Additionally, the attitudes of being proactive, persistent, and accountable will enable me to take more initiative, surpass challenges, and work towards achieving successful outcomes in future projects.

Fatma Jamal

Throughout this project, I have learned a lot in many different aspects, including technical skills such as problem-solving, designing and realizing solutions, as well as interpersonal skills such as effective communication and teamwork. I have also learned the importance of adapting to change and acting professionally and ethically. Overall, the new technical skills I have gained during the project experience will be valuable in the long term, as many jobs in the computer science field require strong programming skills. These skills can be used to build software, websites, or applications for clients and help companies make data-driven decisions.

During the final project, I also learned a few shortcomings that need to be avoided in future projects, including time management, lack of communication, and insufficient testing. By avoiding these, project outcomes could improve significantly, meeting requirements and deadlines successfully. In order to keep a project successful, it is essential to focus on collaboration, teamwork, adaptability, and flexibility. These skills will be valuable in future careers and projects, as the computer science field is constantly changing and evolving.

Fatma Al Kuwari

The senior project was a great opportunity for me to apply my academic knowledge and skills to a real-world project. And some of the lessons I learnt are:

- **Time management:** The senior project had a tight deadline which required effective time management skills. I learnt how to prioritize tasks, set achievable goals, and meet deadlines.
- **Project planning:** I learnt some project management skills such as creating a project plan and managing project risks.
- **Problem-solving:** I learnt how to Identify problems, evaluate potential solutions, and implement effective solutions.

And from what I experienced there are some shortcomings that should be avoided in each future project such as:

- **Inadequate planning:** poor planning can lead to delays and ultimately project failure. It is very important to define clear goals, requirements, timelines and milestones.
- **Lack of communication:** effective communication is essential for successful software development project. Without good communication between team members misunderstandings can arise, requirements can be misinterpreted, and progress can be slowed.
- **Feature creep:** adding too many features can lea to complexity, longer development cycles, and ultimately a product that does not meet the needs. It is important to focus on the core features and prioritize them based on the needs.

Overall, the senior project provided me valuable learning experience that prepared me for future careers, and I learnt valuable skills such as time management, project planning and problem solving which are essential for success in any field.

References

1. *UDST libguides: Learning Commons: Equipment Loan & Software*. Equipment Loan & Software - Learning Commons - UDST LibGuides at University of Doha for Science and Technology. (n.d.). Retrieved from https://library.udst.edu.qa/learningcommons/dmc/equipment
2. *For laboratories that want to get more research done*. Clustermarket. (n.d.). Retrieved from https://clustermarket.com/
3. *11th annual International Conference on Industrial Engineering and Operations Management Singapore, March 7-11, 2021*. singapore 2021. (2021). from http://ieomsociety.org/singapore2021/
4. Lim, M. C., Obispo, J. K. T., Ong, S. L. R., & Tan, G. S. L. (n.d.). *Automated borrowing system for laboratory equipment using RFID technology*. Animo Repository. Retrieved from https://animorepository.dlsu.edu.ph/etd_bachelors/14681/
5. *Laboratory equipment borrowing system of LORMA Colleges Skills Development Institute Free essay example*. StudyMoose. (2021, February 22). Retrieved from https://studymoose.com/laboratory-equipment-borrowing-system-of-lorma-colleges-skills-development-institute-essay
6. *Track your loaner lab equipment using an App - tool tracking software*. (n.d.). Retrieved from https://gocodes.com/track-your-loaner-lab-equipment-using-an-app/
7. Lgalletly@middlebury.edu. (2022, May 9). *Equipment borrowing*. Middlebury Libraries. Retrieved from https://www.middlebury.edu/library/services/borrowing/equipment-borrowing
8. Lorensbergs. (n.d.). *Equipment checkout software - what is it and why use it?* Lorensbergs - Connect2 software. Retrieved from https://www.connect2software.com/blog/equipment-checkout-software/
9. *Home | Qatar National Library - QNL*. (n.d.). Retrieved from https://www.qnl.qa/en
10. *Library: Qatar University*. Library | Qatar University. (n.d.). Retrieved from http://www.qu.edu.qa/library
11. Zeil, S. J. (2015, October 19). Software development process models. Retrieved from https://www.cs.odu.edu/~zeil/cs350/f15/Public/processModels/

12. *Download visual paradigm*. Download Visual Paradigm. Try it FREE. (n.d.). Retrieved from https://www.visual-paradigm.com/download/
13. *Layered architecture style by roast duck - university of waterloo*. (n.d.). Retrieved, from https://student.cs.uwaterloo.ca/~cs446/1171/Arch_Design_Activity/Layered.pdf
14. Richards, M. (n.d.). *Software architecture patterns*. O'Reilly Online Learning. Retrieved from https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html
15. Miller, B. (2017, November 2). *6 pros and cons of Object Oriented Programming*. Green Garage. Retrieved from https://greengarageblog.org/6-pros-and-cons-of-object-oriented-programming
16. *Pipe and filter architectural style - university of waterloo*. (n.d.). Retrieved from https://student.cs.uwaterloo.ca/~cs446/1171/Arch_Design_Activity/PipeFilter.pdf
17. *Let's build from here*. GitHub. (n.d.). Retrieved from https://github.com/
18. *The code affirms an obligation of computing professionals to use their skills for the benefit of society.* Code of Ethics. (n.d.). Retrieved from https://www.acm.org/code-of-ethics
19. *IEEE code of Ethics*. IEEE. (n.d.). Retrieved from https://www.ieee.org/about/corporate/governance/p7-8.html
20. Google. (n.d.). *FireBase* . Google. Retrieved from https://firebase.google.com/
21. *Build apps for any screen*. Flutter. (n.d.). Retrieved from https://flutter.dev/
22. Microsoft. (2021, November 3). *Visual studio code - code editing. redefined*. RSS. Retrieved from https://code.visualstudio.com/
23. *Citation generator*. Citation Machine, a Chegg service. (n.d.). Retrieved from https://www.citationmachine.net/apa
24. *Clickup University*. ClickUp University. (n.d.). Retrieved from https://university.clickup.com/
25. Google. (n.d.). *Google Forms* . Google forms: Online form creator | google workspace. Retrieved from https://www.google.com/forms/about/
26. Homebrew. (n.d.). Retrieved from https://brew.sh/
27. *Directus: The Modern Data Stack, democratized*. RSS. (n.d.). Retrieved from https://directus.io/

28. Inc., A. (n.d.). *Xcode 14 overview*. Apple Developer. Retrieved from https://developer.apple.com/xcode/
29. *Download Android Studio & App Tools*. Android Developers. (n.d.). Retrieved from https://developer.android.com/studio?gclid=Cj0KCQiAmaibBhCAARIsAKUlaKSdkauGQiXgfcjg5PAf80MuQ6tm9CgVSunNbg-_fDnj3hLzaOUu4kMaAi9AEALw_wcB&gclsrc=aw.ds
30. *Microsoft project*. Project Management Software. (n.d.). Retrieved from https://www.microsoft.com/en-us/microsoft-365/project/project-management-software
31. Google. (n.d.). *Senior survey - students*. Google. Retrieved from https://docs.google.com/forms/d/e/1FAIpQLSdKuCJdNS_v_Hd_ShYVs2wY0yWUB6B7chYhaVqlYgh-3mundQ/viewform
32. *Diagrams.net - free flowchart maker and diagrams online*. Flowchart Maker & Online Diagram Software. (n.d.). Retrieved from https://app.diagrams.net/#G1XKhpKXKNcAe8kIinLR7ggcFXWzjtsN2U
33. *Online gantt chart maker for Project Planning*. GanttPRO. (n.d.). Retrieved from https://ganttpro.com/
34. *Modern Mobile app kit (flexible duration)*. Renderforest. (n.d.). https://www.renderforest.com/template/modern-mobile-app-kit
35. *Draw.io - free flowchart maker and diagrams online*. Flowchart Maker & Online Diagram Software. (n.d.). https://app.diagrams.net/

Appendix A – Use cases specification

Use case Id: UC**01**	Sign In
Brief Description	User signs in by using their QU email.
Primary actors	Head of department, admin, general user.
Preconditions: - User has already signed up.	
Post-conditions: - User signed in and can presently utilize highlights that require sign in.	
Main Success Scenario:	
Actor Action	**System Response**
1. User clicks sign in.	2. Display the sign in the user interface.
3. User enters their credentials: username and password.	4. Check the validity.
	5. Update and display the user interface page of the app based on user role
Alternative flows: 4.a. if the credentials are invalid, show an error message. 4.b. if credentials do not exist, pop up a sign-up option for the user.	
Special Requirements:	

Use case Id: UC**02**	Sign Up
Brief Description	User signs up and create an account based on their role.
Primary actors	Head of department, Admin, General User.
Preconditions:	
Post-conditions: - User credentials were created.	

Main Success Scenario:	
Actor Action	**System Response**
1. User clicks sign up.	2. Display the sign-up user interface page and role selection list for the user.
3. User selects the role from the dropdown.	4. Display the sign-up user interface that takes user inputs.
5. User enters the inputs for sign up and confirms.	6. Create new account for the user.
Alternative flows: 6.a. if a required field is missing, ask the user to enter the input in the field.	
Special Requirements:	

Use case Id: UC**03**	Sign Out
Brief Description	User signs out
Primary actors	Head of department, admin, general user.
Preconditions: - User has already signed in.	
Post-conditions: - User was not authenticated in the system any longer and has no access to authorized data.	
Main Success Scenario:	
Actor Action	**System Response**
1. User clicks sign out.	2. Display the sign out confirmation dialog.
3. User clicks in confirm.	4. Sign the user out.
	5. Navigate back to sign-in screen.
Alternative flows: 3.a. if the user taps on “Cancel” the flow of sign out is aborted.	

Special Requirements:

Use case Id: UC**04**	Reset Password
Brief Description	User reset password
Primary actors	Head of department, admin, general user.
Preconditions: Users must already be registered in the system.	
Post-conditions: User's password was changed.	
Main Success Scenario: User's password is changed using which user can sign-in to the system	
Actor Action	**System Response**
1. User clicks on the "Forgot Password" button.	2. Navigate to a form which asks for user's email.
3. User enters their email which was used to register in the system and clicks on "Submit" button (See 3.a.).	4. Send user an email with link.
5. User opens the link in the email.	6. Navigate the user to a form which asks the user for a new password.
7. User enters a new password and clicks on the "Submit" button.	8. Set the new password and then display a message indicating successful update of password.
Alternative flows: 3.a. If the user is not already registered in the system and the email does not exist in the system, display an error.	
Special Requirements:	

Use case Id: UC**05**	Process Requests
Brief Description	Users can act upon active requests for items, by accepting or declining them. Based on the action type, the request proceeds to the role next in turn or gets cancelled.
Primary actors	Admin, Faculty, Head of Department
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: Faculty, Admin and be logged in	
Post-conditions: Request status was updated	
Main Success Scenario: Request status was updated	
Actor Action	**System Response**
1. User navigates to the list of requests to view them (See UC26: "View All Requests")	2. Display all active requests.
3. User clicks on "Approve" or "Decline" button	4. Process the request and based on the action type and based on current role of the user, update the state of the request.
Alternative flows:	
Special Requirements:	

Use case Id: UC**06**	Add Comment
Brief Description	Users can add a comment to a request for an item.
Primary actors	Faculty, admin, head of department.

Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: Faculty, Admin and be logged in.	
Post-conditions: Comment was added to the request.	
Main Success Scenario: Comment is added to the request.	
Actor Action	**System Response**
1. User navigates to the list of requests to view them (See UC26: “View All Requests”).	2. Display all active requests.
3. User clicks on “Add Comment” button.	4. Display a “Comment” field to the user.
5. User enters their comment.	6. Save the comment in the request.
Alternative flows:	
Special Requirements:	

Use case Id: UC**07**	View All Bookings
Brief Description	User can browse all active bookings in the system.
Primary actors	Faculty, Admin.
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: Faculty, Admin and be logged in	
Post-conditions: User viewed all the bookings.	
Main Success Scenario: User views all the bookings.	
Actor Action	**System Response**

1. User clicks on “View all Bookings”.	2. Display all bookings.
Alternative flows:	
Special Requirements:	

Use case Id: UC**08**	View Booking Details
Brief Description	User can view more details of a specific booking by clicking on it.
Primary actors	Faculty, Admin.
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: Faculty, Admin, and be logged in.	
Post-conditions: User was able to view the details related to a specific booking.	
Main Success Scenario: User is able to view the details related to a specific booking.	
Actor Action	**System Response**
1. User clicks on a booking while being in “View All Bookings” page.	2. Display the details of the booking that the user clicked on.
Alternative flows:	
Special Requirements:	

Use case Id: UC**09**	Scan Student Card
Brief Description	User can scan the ID of a student and view all details related to the student that are stored in the system, such as Profile, bookings made, etc. In case, the details do not exist in the system, which can be

	possible when the database of the system is not in sync with the master database of the university.
Primary actors	Faculty.
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: Faculty, Admin, and be logged in.	
Post-conditions: User was able to see all the details related to the student.	
Main Success Scenario: User is able to see all the details related to the student.	
Actor Action	**System Response**
1. User opens Barcode scanner in the mobile app.	2. Display the scanner and is ready to scan.
3. User holds the barcode of the Student ID card in front of the device's camera.	4. Retrieve the student ID number by scanning the barcode.
	5. Look up the student details using the student ID number (See 5.a. for alternative flow).
	6. Display the details of the student.
Alternative flows: 5.a. If the student details do not exist in the system, display a message indicating that the details do not exist for that student.	
Special Requirements:	

Use case Id: UC**10**	View Student Details
Brief Description	View all information about the student.
Primary actors	Faculty, admin.
Preconditions: User must be registered in the system.	

Post-conditions: User was able to see all the information about the student.	
Main Success Scenario:	
Actor Action	**System Response**
1. User selects the student that they want to see all details about	2. Display the details of the student.
Alternative flows:	
Special Requirements:	

Use case Id: UC**11**	View Request Status
Brief Description	Users can view the status of the requests for new items they made, whether it is “Approved” or “Declined” or if there are any comments.
Primary actors	General User
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: General User, and be logged in**.**	
Post-conditions: User viewed status of the request as well as any comments	
Main Success Scenario: User views status of the request as well as any comments	
Actor Action	**System Response**
1. User navigates to the “Requests” page.	2. Display all requests made by the user in the past.
3. To view any comment user clicks on “Show Comments” button (See alternative flow 3.a).	4. Display all comments made by different users, along with the time when the comment was made.

Alternative flows: 3.a. Incase comments do not exist for the request, display a message indicating that there are no comments yet.
Special Requirements:

Use case Id: UC**12**	View Booking History
Brief Description	User views history of items that were booked by them in the past.
Primary actors	Faculty, admin.
Preconditions: User has already booked an item.	
Post-conditions: User viewed the history of all items that were booked by them.	
Main Success Scenario: User views the history of all items that were booked.	
Actor Action	**System Response**
1. User navigates to the "History" page.	2. Display all bookings that were made by the user in the past.
Alternative flows:	
Special Requirements:	

Use case Id: UC**13**	Request New Item
Brief Description	Users request for an Item that does not exist in the campus facilities or item is overbooked. They enter all details as well as cost, if known. This request then is reviewed by other actors as specified in other use cases.
Primary actors	General User

Preconditions: User is registered in the system and logged in	
Post-conditions: User has registered a new item request	
Main Success Scenario: Active Item Request is registered under the user. It is now viewable by the concerned users	
Actor Action	**System Response**
1.User enters item details	
2.User clicks Submit	3. Display confirmation message
4. User confirms (See alternative flow 4.a.)	5. Save the request
	6. Display success message
Alternative flows: 4.a. If the user cancels the confirmation message the flow is aborted.	
Special Requirements:	

Use case Id: UC**14**	View Item Details
Brief Description	User can view details about a specific item.
Primary actors	General User, Admin.
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: General User, Admin, and be logged in.	
Post-conditions: User is able to view all item details.	
Main Success Scenario: User was able to view all item details.	

Actor Action	System Response
1. User clicks on an item to view the details.	2. Display all details about the item.
Alternative flows:	
Special Requirements:	

Use case Id: UC**15**	Search item
Brief Description	User can search for an item by ID or name.
Primary actors	General User.
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: General User, and be logged in.	
Post-conditions: User was able to view the item if it exists in the system.	
Main Success Scenario: User is able to view the item if it exists in the system.	
Actor Action	**System Response**
1. User clicks on the search bar.	2. Enable focus on the text field in search bar and allow the user to type.
3. User enters the ID or name of the item that they are searching for (See 3.a. for alternative flow).	4. Display the item in the list.
Alternative flows: 3.a. If the item does not exist in the system, a message indicating that the item was not found is displayed.	
Special Requirements:	

Use case Id: UC**16**	Book an Item
Brief Description	Users book an item for a certain period of time. During this time period it is to be considered that the item is in possession of the user who made the booking. The system updates the list of items in the user account.
Primary actors	General User, Student, Faculty
Preconditions: 1. User was registered in the system and logged in 2. The item was not already booked by someone else during the time period	
Post-conditions: 1. User has a new booking under them with details about the time period and item booked The item booked is not available anymore for other users during the time period	
Main Success Scenario: The item is successfully booked for the selected time period and not available anymore for other users to book during that period	
Actor Action	**System Response**
1. User scans the barcode of the item	2. Display the item details
3. User selects the time period of booking	4. Check if the item is available during the time period
	5. If item is available, enable the greyed out "Continue" button
6. User accepts agreement (See extended Use Case "Accept Agreement")	
	7. Display confirmation message for the booking
Alternative flows:	

4.a. If the item is not available, disallow the user to proceed and provides the option to select a different time period.
Special Requirements:

Use case Id: UC**17**	Accept agreement
Brief Description	Users accept agreement while booking an item.
Primary actors	Faculty, Student
Preconditions: User is booking an item.	
Post-conditions: Users accepted the agreement and successfully book an item.	
Main Success Scenario:	
Actor Action	**System Response**
1. User ticks all the check boxes.	2. Enable the greyed out "Agree" button
3. User taps on "Agree" button	4. Navigate to the next step of booking item (See Use Case UC03)
Alternative flows: 3.a. if users do not accept the agreement, remind the user to accept the agreement.	
Special Requirements:	

Use case Id: UC**18**	Return an item
Brief Description	Return an item that has been borrowed from the lab.
Primary actors	General user, Faculty, Student.
Preconditions: The item has already borrowed by the signed in user.	
Post-conditions: The item was made available.	

Main Success Scenario: The return item was accepted.	
Actor Action	**System Response**
1. User scans the item.	2. Retrieve the item details.
	3. Update the user account.
	4. Make the item available.
	5. Decrement the total items borrowed of the user by 1.
Alternative flows:	
Special Requirements:	

Use case Id: UC**19**	Cancel booking	
Brief Description	User can cancel the booking of an item.	
Primary actors	General User.	
Preconditions: - User was registered in the system and logged in - User had already booked an item.		
Post-conditions: Booking of an item during the period has been cancelled.		
Main Success Scenario: Booking of an item during the period has been cancelled.		
Actor Action		**System Response**
1. User selects the item that is booked.		
2. User clicks on “Cancel Booking”.		3. Prompt the user to confirm cancellation.
4. User clicks on “Confirm” button.		5. Cancel the booking.
		6. Change item status from “Reserved” to “Available” during the period selected in the booking.

Alternative flows:
Special Requirements:

Use case Id: UC**20**	Create New Item
Brief Description	Admin creates new items in the system by entering the relevant item details.
Primary actors	Admin
Preconditions: Item details are collected.	
Post-conditions: A new item was created by admin which can be viewed in respective lists	
Main Success Scenario: New item is available in the application and ready to be used/booked.	
Actor Action	**System Response**
1. Admin clicks on “Create New Item”.	2. Display a new empty page with empty information boxes.
3. Fill item details and uploads the image.	
3. Admin clicks “create”.	4. Create the item and make it available to be booked.
Alternative flows: 3.a. If admin did not fill all the empty places, remind the admin to fill them all.	
Special Requirements:	

Use case Id: UC**21**	Edit Item

<table>
<tr><td><b>Brief Description</b></td><td>User updates details of an item.</td></tr>
<tr><td><b>Primary actors</b></td><td>Faculty, Admin.</td></tr>
<tr><td colspan="2"><b>Preconditions:</b>
Faculty members will choose the item they want to edit.</td></tr>
<tr><td colspan="2"><b>Post-conditions:</b>
Information of a specific item was edited.</td></tr>
<tr><td colspan="2"><b>Main Success Scenario:</b>
Information of a specific item was edited.</td></tr>
<tr><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td>1. Faculty member clicks “Edit” button</td><td>2. Enable the faculty member to make changes.</td></tr>
<tr><td>3. Make the changes and edit the information</td><td></td></tr>
<tr><td>4. Click “Confirm” button to save the changes.</td><td>5. Save the changes and successfully edit the information.</td></tr>
<tr><td colspan="2"><b>Alternative flows:</b></td></tr>
<tr><td colspan="2"><b>Special Requirements:</b></td></tr>
</table>

<table>
<tr><td><b>Use case Id:</b>
UC<b>22</b></td><td>Archive Items</td></tr>
<tr><td><b>Brief Description</b></td><td>When the equipment in labs may no longer be in use they can be archived. This archiving action will hide the items from all the lists but still be displayed in histories of all users if they performed a transaction with it.</td></tr>
<tr><td><b>Primary actors</b></td><td>Admin.</td></tr>
<tr><td colspan="2"><b>Preconditions:</b>
User must be registered in the system, belong to a role that has permission to perform this action: Faculty, Admin, and be logged in.</td></tr>
<tr><td colspan="2"><b>Post-conditions:</b>
The item was removed from all lists and is not visible anymore</td></tr>
</table>

Main Success Scenario: The item can no longer be booked and is not visible in any searches.	
Actor Action	**System Response**
1. User clicks on “Archive Item” in the context menu of an item.	2. System displays a confirmation dialog.
3. User clicks on “Confirm” (See alternative flow 3.a.).	4. System archives the item.
Alternative flows: 3.a. If user clicks on cancel the use case aborts.	
Special Requirements:	

Use case Id: UC**23**	Generate Reports
Brief Description	Admin view report of all the past booking transactions.
Primary actors	Admin
Preconditions: Admin requests a report of past booking transactions.	
Post-conditions: Admin viewed a full report that included all past bookings.	
Main Success Scenario: Admin views all past bookings in one report.	
Actor Action	**System Response**
1. Admin clicks on “Generate Report”.	2. Retrieve all information from past bookings.
	3. Generate the report for downloading.
4. Download the report.	
Alternative flows:	
Special Requirements:	

<table>
<tr><td><b>Use case Id:</b>
UC<b>24</b></td><td colspan="2">View profile</td></tr>
<tr><td><b>Brief Description</b></td><td colspan="2">Users can view their profiles to check their personal details such as name, ID, items booked by them, etc.</td></tr>
<tr><td><b>Primary actors</b></td><td colspan="2">General User, Admin, Head of Departments.</td></tr>
<tr><td colspan="3"><b>Preconditions:</b>
User must be registered in the system, belong to a role that has permission to perform this action: Faculty, Admin, and be logged in.</td></tr>
<tr><td colspan="3"><b>Post-conditions:</b>
User was able to view all their personal details.</td></tr>
<tr><td colspan="3"><b>Main Success Scenario:</b>
User is able to view all their personal details.</td></tr>
<tr><td colspan="2"><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td colspan="2">1. User navigates to “User Profile”.</td><td>2. System displays the profile for the user that is logged in.</td></tr>
<tr><td colspan="3"><b>Alternative flows:</b></td></tr>
<tr><td colspan="3"><b>Special Requirements:</b></td></tr>
</table>

<table>
<tr><td><b>Use case Id:</b>
UC<b>25</b></td><td>Update Profile</td></tr>
<tr><td><b>Brief Description</b></td><td>Users can view their profiles to check their personal details such as name, ID, items booked by them, etc.</td></tr>
<tr><td><b>Primary actors</b></td><td>General user, admin, head of department.</td></tr>
<tr><td colspan="2"><b>Preconditions:</b>
User must be registered in the system, belong to a role that has permission to perform this action: General user, admin, head of department, and be logged in.</td></tr>
<tr><td colspan="2"><b>Post-conditions:</b>
The profile was updated with the details entered by the user.</td></tr>
<tr><td colspan="2"><b>Main Success Scenario:</b></td></tr>
</table>

The profile is updated with the details entered by the user.	
Actor Action	**System Response**
1. User navigates to “User Profile”.	2. Display the profile for the user that is logged in.
3. User clicks on “Edit” button which is indicated by either text or an icon.	4. Display editable fields which contain information that were already saved in the system.
5. User edits the information that they desire to update.	
6. User clicks on “Submit’ button.	7. Update all the details in the user’s profile (See 7.a.).
Alternative flows: 7.a. If the information is not in valid format, display an error and allow the user to edit the information and retry.	
Special Requirements:	

Use case Id: UC**26**	View all Requests
Brief Description	User can view all requests that may need their approval.
Primary actors	Faculty, Admin, Head of Department.
Preconditions: User must be registered in the system, belong to a role that has permission to perform this action: General user, admin, head of department, and be logged in.	
Post-conditions: User viewed all requests that are due for approval.	
Main Success Scenario: User is able to see all requests that are due for approval.	
Actor Action	**System Response**
1. User clicks on “All Requests” screen.	2. Display all requests that are due for approval for the logged-in user.

Alternative flows:
Special Requirements:

Appendix B – Test cases specification

Use Case: Sign In

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Open the mobile app	The mobile app opens successfully	Mobile app opened	Pass	N/A
Enter username	User can enter their username	Username entered	Pass	N/A
Enter password	System enables login button	Password entered	Pass	N/A
Click on "Sign In" button	System takes user to the home page	Navigated to home page	Pass	N/A

Expected Result: User is successfully signed into the app.

Actual Result: Matches expected result.

Use Case: Sign Up

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Open the mobile app	The mobile app opens successfully	Mobile app opened	Pass	N/A
Click on “Don’t have an	System displays three fields	Clicked on "Sign Up" button	Pass	N/A

account?" button	for name, email and password			
Enter name	Hide the name hint	Name entered	Pass	N/A
Enter email	Hide the email hint	Username entered	Pass	N/A
Enter password	Hide the password hint	Password entered	Pass	N/A
Click on ">" button	System clears fields and displays sign in screen	Displayed sign in screen	Pass	N/A

Expected Result: User is successfully signed up and registered in the system.

Actual Result: Matches expected result.

Use Case: Sign Out

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Open the mobile app	The mobile app opens successfully	Mobile app opened	Pass	N/A
Switch to last tab	Displays last tab	Displayed last tab	Pass	N/A
Click on "Log Out" button	Navigates back to login page	Navigated back to login page	Pass	N/A

Expected Result: User is successfully signed out of the app.

Actual Result: Matches expected result.

Use Case: Reset Password

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Open the web portal	Web portal opens successfully	Web portal opened successfully	Pass	N/A
Click on "Forgot Password"	User can click on the "Forgot Password" link	Clicked on "Forgot Password"	Pass	N/A
Enter email	User can enter their email	Entered email	Pass	N/A
Click on the link in the received email	User can click on the reset password link in the received email	Clicked on reset password link	Pass	N/A
Enter new password	User can enter their new password	Entered new password	Pass	N/A
Click confirm	Password is reset and the user is able to sign into the app with	Password reset successfully	Pass	N/A

	the new password			

Expected Result: Upon clicking the confirm button, the password is reset, and the user can sign into the app with the new password.

Actual Result: Matches expected result.

Use Case: Process Request

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Pick a request or booking	User can select a request or booking from the list	Request/booking selected	Pass	N/A
Swipe to Approve	User can swipe the request/booking to approve it	Request/booking approved	Pass	N/A
Click "x" button to Reject	User can click the "x" button to reject the request/booking	Request/booking rejected	Pass	N/A

Expected Result: Upon approving or rejecting the request/booking, it disappears from the list, and a notification is sent to the user linked with the request/booking.

Actual Result: Matches expected result.

Use Case: Add Comment

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Navigate to the last tab	User can navigate to the last tab	Navigated to last tab	Pass	N/A
Tap on "Request New Item"	User can tap on the "Request New Item" button	Tapped on "Request New Item"	Pass	N/A
Enter comment	User can enter a comment in the text field	Entered comment	Pass	N/A

Expected Result: Upon creating the request, the comment is visible to the Head of Department or Faculty.

Actual Result: Matches expected result.

Use Case: View All Bookings

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Navigate to the second tab	User can navigate to	Navigated to second tab	Pass	N/A

	the second tab			
View bookings	User can view all bookings in the list	Viewed bookings	Pass	N/A
Tap on any booking to view more details	User can tap on any booking to view more details	Displayed all booking details	Pass	N/A

Expected Result: User can see all bookings and their respective statuses from the same screen. They can also view more details by tapping on any booking.

Actual Result: Matches expected result.

Use Case: View Booking Details

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Navigate to the second tab	User can navigate to the second tab	Navigated to second tab	Pass	N/A
View bookings	User can view all bookings in the list	Viewed bookings	Pass	N/A

Tap on any booking to view more details	User can tap on any booking to view more details	Displayed all booking details	Pass	N/A

Expected Result: User can see all booking details on which they tapped.

Actual Result: Matches expected result.

Use Case: Scan Student Card

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Click on "Scan Barcode" button	Display barcode scanner screen	Displayed barcode scanner screen	Pass	N/A
Scan barcode on a student ID card	Display student details	Displayed student details	Pass	N/A

Expected Result: Upon scanning the card, student details stored in the system are displayed.

Actual Result: Matches expected result.

Use Case: View Request Status

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Students:				
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Navigate to the Notifications screen by clicking on the Bell icon	Navigate to notifications screen	Navigated to notifications screen	Pass	N/A
Faculty:				
Sign in to the portal	Display home page	Displayed home page	Pass	N/A
Navigate to "Requests"	Display “Requests" tab	Displayed "Requests" tab	Pass	N/A

Expected Result: Users, both students and faculties, can view the status of the requests made by them.

Actual Result: Matches expected result.

Use Case: View Booking History

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Navigate to the second tab	User can navigate to	Navigated to second tab	Pass	N/A

	the second tab			
View bookings	User can view all bookings in the list	Viewed bookings	Pass	N/A
Tap on any to view more details	Show booking details	Showed booking details	Pass	N/A

Expected Result: User can see all bookings and their respective statuses from the same screen. They can also view more details by tapping on any booking.

Actual Result: Matches expected result.

Use Case: Request New Item

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Students:				
Sign in to the mobile app	Display home page	Displayed home page	Pass	N/A
Navigate to the last tab	User can navigate to the last tab	Navigated to last tab	Pass	N/A
Click on "Request New Item"	User can click on the "Request New Item" button	Clicked on "Request New Item"	Pass	N/A
Enter details	User can enter the required	Entered details	Pass	N/A

	details for the new item request			
Click on submit button	User can click on the submit button to create the request	Submitted the request	Pass	N/A
Faculty:				
Sign in to the portal	Display home page	Displayed home page	Pass	N/A
Navigate to "Requests"	User can navigate to the "Requests" tab	Navigated to "Requests" tab	Pass	N/A
Click on "+" button	User can click on the "+" button to create a new request	Clicked on "+" button	Pass	N/A
Enter details	User can enter the required details for the new item request	Entered details	Pass	N/A
Click on submit button	Display confirmation message and	Displayed confirmation message and	Pass	N/A

	navigate back to list view	navigate back to list view		

Expected Result: Upon clicking the submit button, the request is created, and a confirmation message appears for the user.

Actual Result: Matches expected result.

Use Case: View Item Details

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Student:				
Login to the mobile app	Display home page	Displayed home page	Pass	N/A
Click on any equipment	Display item details	Displayed item details	Pass	N/A
Faculty:				
Login to the portal	Display home page	Displayed home page	Pass	N/A
Click on "Equipments"	Display ”Equipments " tab	Display "Equipments " tab	Pass	N/A
Click on any equipment	User can click on any equipment to view its details	Clicked on equipment	Pass	N/A

Expected Result: User can view all details of the equipment, such as description and availability.

Actual Result: Matches expected result.

Use Case: Search Item

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Student:				
Login to the mobile app	Display home page	Displayed home page	Pass	N/A
Enter equipment name in the search bar	User can enter the equipment name in the search bar	Entered equipment name	Pass	N/A
Faculty:				
Login to the portal	User can log in to the portal	Logged in successfully	Pass	N/A
Click on "Equipments"	User can click on the "Equipments " tab	Clicked on "Equipments " tab	Pass	N/A
Enter equipment name in the search bar	Display search results	Displayed search results	Pass	N/A

Expected Result: The equipment appears in the search results if it exists in the database.

Actual Result: Matches expected result.

Use Case: Book an Item

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**

Login to the mobile app	User can log in to the mobile app	Logged in successfully	Pass	N/A
Tap on desired equipment	User can tap on the desired equipment to view its details	Tapped on equipment	Pass	N/A
Tap "Add to Bag"	Selected equipment is added to the bag	Equipment added to bag	Pass	N/A
Go to the Bag	User can navigate to the Bag section	Navigated to Bag	Pass	N/A
Adjust quantities of the equipments	User can adjust the quantities of the selected equipments in the Bag	Quantities adjusted	Pass	N/A
Tap "Borrow All"	User can tap on "Borrow All" button	Tapped on "Borrow All"	Pass	N/A
Select the desired time period	User can select the desired time period for borrowing	Time period selected	Pass	N/A

Tap Next	User can proceed to the next step	Tapped on Next	Pass	N/A
App checks availability	The app checks the availability of the equipment during the selected time period	Availability checked	Pass	N/A
Click "Continue"	User can click on "Continue" after the availability check	Clicked on "Continue"	Pass	N/A
Enter purpose of borrowing	User can enter the purpose of borrowing	Entered purpose	Pass	N/A
Tick the "I agree" checkbox	User can tick the "I agree" checkbox to accept the terms and conditions	Checkbox ticked and Submit button enabled	Pass	N/A
Tap submit	Create booking and display confirmation	Displayed confirmation	Pass	N/A

Expected Result: The booking is created, and a confirmation message is displayed. The booking can also be viewed in the booking history page.

Actual Result: Matches expected result.

Use Case: Accept Agreement

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Login to the mobile app	User can log in to the mobile app	Logged in successfully	Pass	N/A
Complete all steps of "Book an Item" use case until the last step	User completes all steps of the "Book an Item" use case until the last step	Completed all steps	Pass	N/A
Tick the agreement checkbox	Submit button enabled	Submit button enabled	Pass	N/A

Expected Result: If the user does not agree to the agreement, the borrowing button is grayed out. Otherwise, the user can continue borrowing.

Actual Result: Matches expected result.

Use Case: Return an Item

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment

Login to the mobile app	User can log in to the mobile app	Logged in successfully	Pass	N/A
Go to the Booking History tab	User can navigate to the Booking History tab	Navigated to Booking History	Pass	N/A
Choose the booking	User can choose the booking from the list	Booking chosen	Pass	N/A
Click on "Return All" button	User can click on the "Return All" button to initiate the return process	Clicked on "Return All"	Pass	N/A
Enter confirmation code	Display “Returned” status	Displayed “Returned” status	Pass	N/A

Expected Result: Upon entering the correct confirmation code, the booking status changes to "Returned," and a confirmation message is displayed to the user.

Actual Result: Matches expected result.

Use Case: Cancel Booking

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**

Login to the mobile app	User can log in to the mobile app	Logged in successfully	Pass	N/A
Go to the Booking History tab	User can navigate to the Booking History tab	Navigated to Booking History	Pass	N/A
Choose the booking	User can choose the booking from the list	Booking chosen	Pass	N/A
Click on "Cancel Booking" button	User can click on the "Cancel Booking" button to initiate the cancellation process	Clicked on "Cancel Booking"	Pass	N/A

Expected Result: A confirmation message is displayed, and the status of the booking changes to "Cancelled."

Actual Result: Matches expected result.

Use Case: Create New Item

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Login to the portal	User can log in to the portal	Logged in successfully	Pass	N/A

Go to the "Equipments" tab	User can navigate to the "Equipments" tab	Navigated to "Equipments"	Pass	N/A
Click on "+" button	User can click on the "+" button to create a new item	Clicked on "+" button	Pass	N/A
Enter equipment details	User can enter all equipment details in the form	Entered equipment details	Pass	N/A
Upload equipment picture	User can upload a picture of the equipment	Uploaded equipment picture	Pass	N/A
Click on Save button	Create item and display confirmation message	Item created and confirmation message displayed	Pass	N/A

Expected Result: A confirmation message is displayed to the user, and the user can view the item in the equipment list from both the app and the portal.

Actual Result: Matches expected result.

Use Case: Edit Item

Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Login to the portal	User can log in to the portal	Logged in successfully	Pass	N/A
Go to the "Equipments" tab	User can navigate to the "Equipments" tab	Navigated to "Equipments"	Pass	N/A
Select an equipment	User can select an equipment from the list	Equipment selected	Pass	N/A
Update details of the equipment	User can update the details of the selected equipment	Details updated	Pass	N/A
Click on Save button	Edit item and display confirmation message	Item details updated and confirmation message displayed	Pass	N/A

Expected Result: The user will be able to see the updated details of the equipment reflected in the list, alongside a confirmation message.

Actual Result: Matches expected result.

Use Case: Archive Item

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Login to the portal	User can log in to the portal	Logged in successfully	Pass	N/A
Go to the "Equipments" tab	User can navigate to the "Equipments " tab	Navigated to "Equipments "	Pass	N/A
Select an equipment	User can select an equipment from the list	Equipment selected	Pass	N/A
Update the status of the equipment	User can update the status of the equipment to "Archived"	Status updated to "Archived"	Pass	N/A
Click on Save button	Item is archived	Item is archived	Pass	N/A

Expected Result: A confirmation message will be displayed to the user, and the equipment will not be visible anymore in the mobile application.

Actual Result: Matches expected result.

Use Case: Generate Reports

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Login to the portal	Take the user to the home page	Home page displayed	Pass	N/A
Go to the desired tab	User can navigate to the desired tab (e.g., "Bookings")	Navigated to desired tab	Pass	N/A
Click on "Export Items"	User can click on the "Export Items" option on the right-side bar	Clicked on "Export Items"	Pass	N/A
Choose desired format	User can choose the desired format for the report (e.g., .csv, .json)	Chose desired format	Pass	N/A
Choose fields for the report	User can select the fields to be included in the report	Selected fields for the report	Pass	N/A

Click on "Download" button	File with selected format is downloaded	File with selected format is downloaded	Pass	N/A

Expected Result: A report will be generated in the chosen format and then downloaded in the user's browser's Downloads folder. If it is .csv file it can be opened in Excel.

Actual Result: Matches expected result.

Use Case: View Profile

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Login to the mobile app	User can log in to the mobile app	Logged in successfully	Pass	N/A
Go to the last tab	User can navigate to the last tab	Navigated to last tab	Pass	N/A
Click on user's avatar	View user profile details	Displayed user profile details	Pass	N/A

Expected Result: User will be able to see all profile details.

Actual Result: Matches expected result.

Use Case: Update Profile

Test Step	Expected System Response	Actual Result	Pass/Fail	Comment
Login to the mobile app	Display home page	Displayed home page	Pass	N/A

Go to the last tab	Display the last tab	Navigated to last tab	Pass	N/A
Click on user's avatar	Display User profile details	Displayed User profile details	Pass	N/A
Update details	User can update the desired details in the profile	Details updated	Pass	N/A
Click on the tick icon	Profile details are updated and confirmation message displayed	Profile details updated and confirmation message displayed	Pass	N/A

Expected Result: User profile is updated, and the user can immediately see the new details reflected below the user's avatar.

Actual Result: Matches expected result.

Use Case: View all Requests

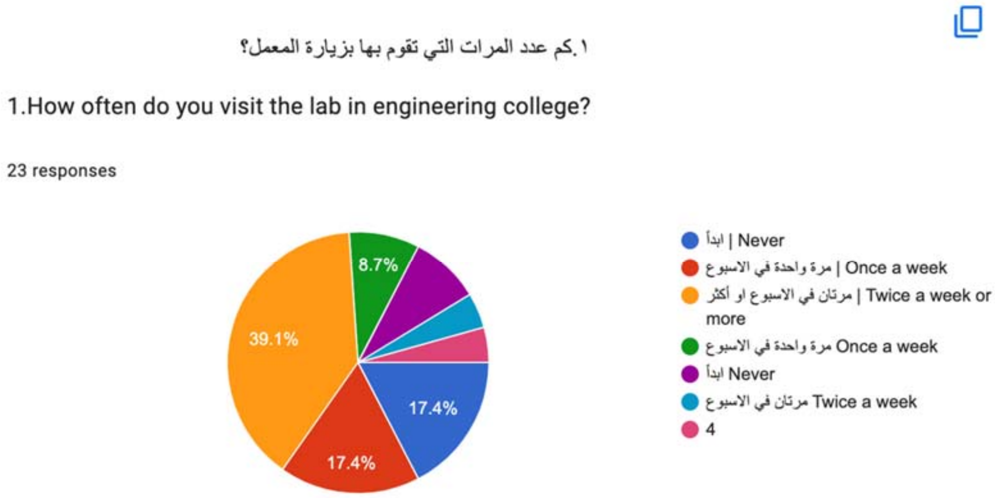
Test Step	**Expected System Response**	**Actual Result**	**Pass/Fail**	**Comment**
Login to the portal	Display home page	Displayed home page	Pass	N/A
Go to "Requests" tab	Display all requests	Displayed all requests	Pass	N/A

Expected Result: User is able to view all requests made by all users of the system

Actual Result: Matches expected result.

Other Appendices:

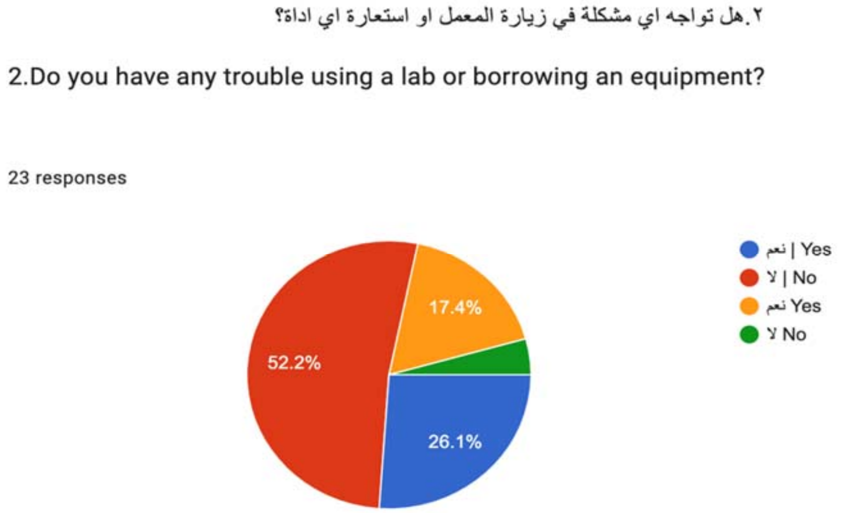
Appendix C – Survey Results



١.كم عدد المرات التي تقوم بها بزيارة المعمل؟

1.How often do you visit the lab in engineering college?

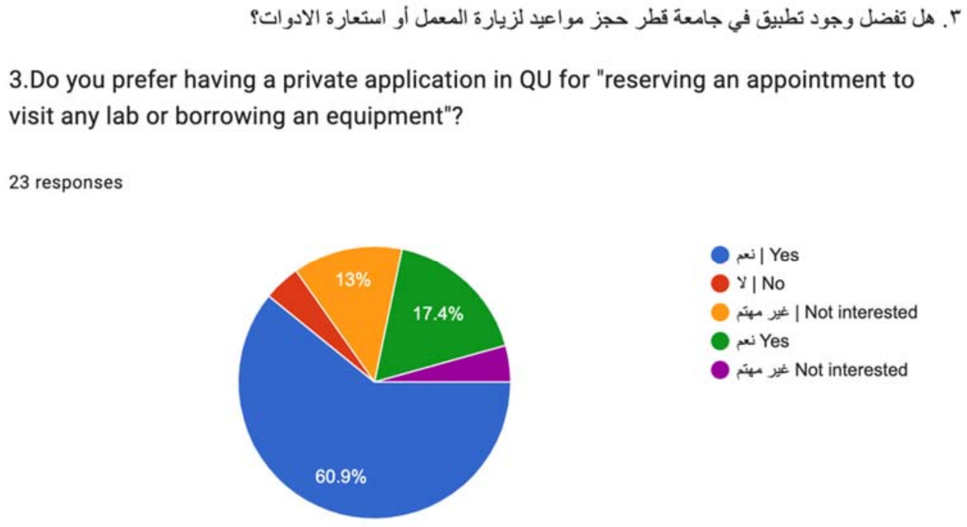
23 responses



٢.هل تواجه اي مشكلة في زيارة المعمل او استعارة اي اداة؟

2.Do you have any trouble using a lab or borrowing an equipment?

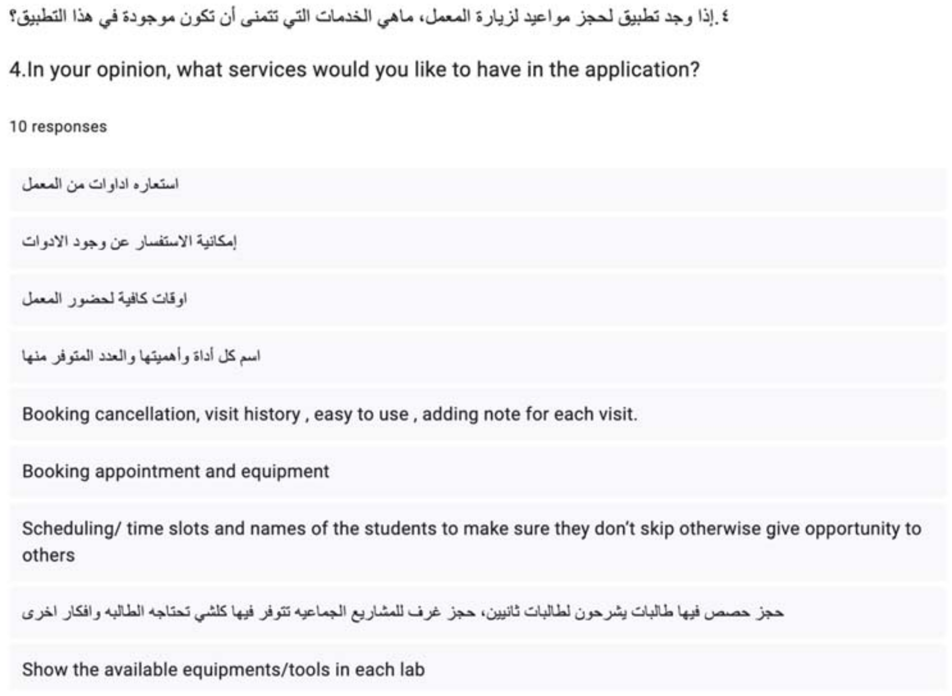
23 responses



٣. هل تفضل وجود تطبيق في جامعة قطر حجز مواعيد لزيارة المعمل أو استعارة الادوات؟

3.Do you prefer having a private application in QU for "reserving an appointment to visit any lab or borrowing an equipment"?

23 responses



٤.إذا وجد تطبيق لحجز مواعيد لزيارة المعمل، ماهي الخدمات التي تتمنى أن تكون موجودة في هذا التطبيق؟

4.In your opinion, what services would you like to have in the application?

10 responses

استعاره اداوات من المعمل

إمكانية الاستفسار عن وجود الادوات

اوقات كافية لحضور المعمل

اسم كل أداة وأهميتها والعدد المتوفر منها

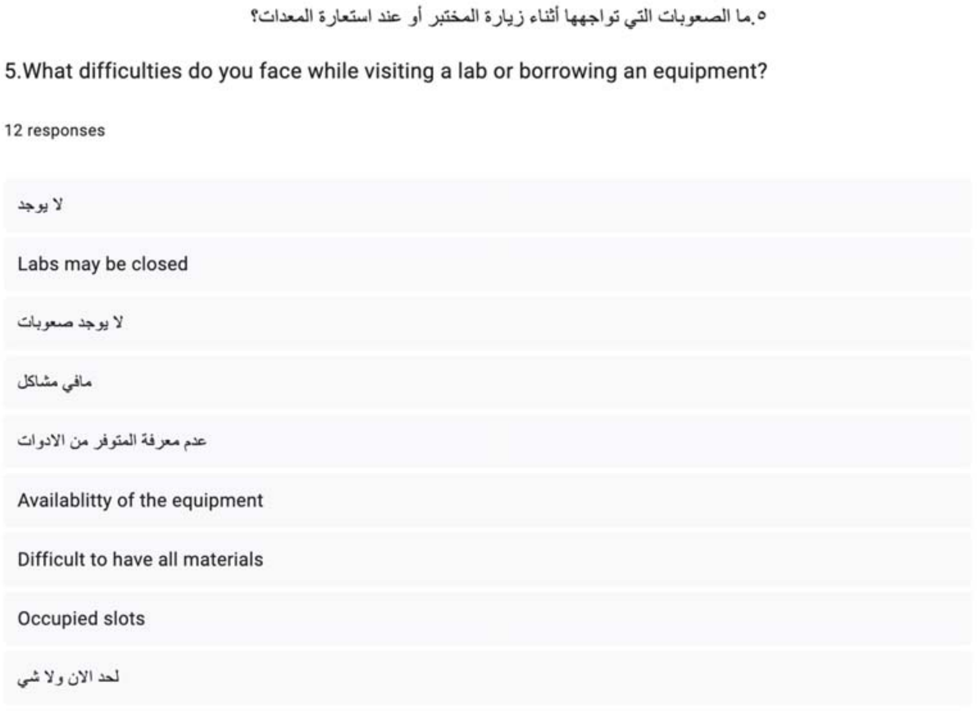
Booking cancellation, visit history , easy to use , adding note for each visit.

Booking appointment and equipment

Scheduling/ time slots and names of the students to make sure they don't skip otherwise give opportunity to others

حجز حصص فيها طالبات يشرحون لطالبات ثانيين، حجز غرف للمشاريع الجماعيه تتوفر فيها كلشي تحتاجه الطالبه وافكار اخرى

Show the available equipments/tools in each lab



٥.ما الصعوبات التي تواجهها أثناء زيارة المختبر أو عند استعارة المعدات؟

5.What difficulties do you face while visiting a lab or borrowing an equipment?

12 responses

لا يوجد

Labs may be closed

لا يوجد صعوبات

مافي مشاكل

عدم معرفة المتوفر من الادوات

Availablitty of the equipment

Difficult to have all materials

Occupied slots

لحد الان ولا شي