



BSCS-S21-014

03-134181-043 MUHAMMAD MUJTABA

03-134181-042 MUHAMMAD MAHAD SHAHBAZ

QueryDesk

In partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Muhammad Tayyab Mir

Department of Computer Sciences
Bahria University, Lahore Campus

December 2021

Certificate



We accept the work contained in the report titled

“QueryDesk”

written by

MUHAMMAD MUJTABA

MUHAMMAD MAHAD SHAHBAZ

as a confirmation to the required standard for the partial fulfillment of the degree of
Bachelor of Science in Computer Science.

Approved by:

Supervisor:

Muhammad Tayyab Mir

(Signature)

December 2021

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134181-043	MUHAMMAD MUJTABA	
03-134181-042	MUHAMMAD MAHAD SHAHBAZ	

Date : December 30, 2021

Specially dedicated to
my beloved grandmother, grandfather, mother, and father
(MUHAMMAD MUJTABA)
my beloved grandmother, grandfather, mother, and father
(MUHAMMAD MAHAD SHAHBAZ)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our supervisor, Muhammad Tayyab Mir for his invaluable advice, guidance, and his enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parents and friends who had helped and encouraged me.

**MUHAMMAD MUJTABA
MUHAMMAD MAHAD SHAHBAZ**

Project Title**QueryDesk****ABSTRACT**

Application processing is a tiring and time taking process that involves many administrative individuals along with the applicants. The application process requires the timely availability of all authorized personals in the process. Keeping a record of all application take too much file work and human resources. Every educational institute receives a bundle of applications from students daily. Some of the applications may require immediate action, but an outdated application process can stall that application at any level.

QueryDesk offers a mobile application for students to submit their applications using QueryDesk. The main benefit is a quick response from authorities and status available to all members. In the manual process replying to each application for a relative person takes too much time as vetting and evaluating the application is mandatory. QueryDesk can fully digitize your application process from applying to the final stage. QueryDesk collects all documents on one platform and makes them accessible remotely. By QueryDesk you can eliminate repetitive work and save up your time. QueryDesk keeps a record of student application and generates the reports for analysis.

TABLE OF CONTENTS

DECLARATION	ii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF SYMBOLS / ABBREVIATIONS	xiii

CHAPTERS

1	INTRODUCTION	14
	1.1 Background	14
	1.2 Problem Statements	15
	1.3 Aims and Objectives	15
	1.4 Scope of Project	15
2	SOFTWARE REQUIREMENT SPECIFICATION	17
	2.1 User Classes and Characteristics	17
	2.2 Operating Environment	18
	2.3 Development System	19
	2.3.1 Design and Implementation Constraints	19
	2.4 Assumptions and Dependencies	20
	2.5 External Interface Requirements	20
	2.5.1 User Interfaces	20
	2.6 Software Interfaces	21

2.7	System Use Cases	22
2.7.1	Sign-Up as a Student (U1)	23
2.7.2	Sign-In as Student (U2)	24
2.7.3	Sign-In as Faculty Member (U3)	25
2.7.4	Sign-In as Administrator (U4)	26
2.7.5	New Application Submission (U5)	27
2.7.6	Approve Application (U6)	28
2.7.7	Reject Application (U7)	29
2.7.8	Application Status (U8)	30
2.8	Other Non-functional Requirements	31
2.8.1	Performance Requirements	31
2.8.2	Safety Requirements	31
2.8.3	Security Requirements	31
2.8.4	Software Quality Attributes	31
3	DESIGN AND METHODOLOGY	33
3.1	Features-Driven development	33
3.2	Use-case Description	34
3.2.1	Sign up	34
3.2.2	Sign In	35
3.2.3	Add Users	36
3.2.4	Modify Users	37
3.2.5	Sign Out	38
3.2.6	Generate Reports	39
3.2.7	Check Status	40
3.2.8	Approve application	41
3.2.9	Reject application	42
3.2.10	History	43
3.3	Sequence Diagrams	44
3.3.1	Sign in	44
3.3.2	Sign up	44
3.3.3	Check status	45
3.3.4	Report generation	45

3.3.5	Approve application	46
3.3.6	Reject application	46
3.3.7	Add user	47
3.3.8	Modify user	47
3.3.9	Mark Application	48
3.4	Class Diagram	48
3.5	ERD Diagram	49
4	IMPLEMENTATION	50
4.1	Languages used for implementation	50
4.1.1	Dart	50
4.2	Framework:	50
4.2.1	Flutter	50
4.3	Tools	51
4.3.1	VS code	51
4.3.2	Android Studio	51
4.3.3	Android Emulator	51
4.3.4	Abode XD	51
4.4	Online services	52
4.4.1	Firebase	52
4.4.2	Cloud Firestore	52
4.4.3	Firebase Cloud Messaging	52
5	RESULTS AND DISCUSSIONS (or USER MANUAL)	53
5.1	Introduction	53
5.2	Sign In and Sing Up	55
5.3	Student Dashboard	56
5.4	New Application Submission	57
5.5	Application Status	58
5.7	Profile Tab	59
5.8	Authorized Dashboard	60
5.9	Pending Application	61
5.10	Application details	62

5.11	Reports	65
5.12	History	66
5.13	Push notification	67
6	CONCLUSION AND RECOMMENDATIONS	68
6.1	Project Achievements	68
6.2	Future Work	68
6.3	Implementation Issues and Challenges	69
6.4	Conclusion	69
	REFERENCES	70

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1	Software Interfaces	21
Table 2.2	Sign-Up as a Student	23
Table 2.3	Sign-In as a Student	24
Table 2.4	Sign-In as a Faculty Member	25
Table 2.5	Sign-In as Administrator	26
Table 2.6	New Application Submission	27
Table 2.7	Accept Application	28
Table 2.8	Reject Application	29
Table 2.9	Application Status	30
Table 3.1	Sign up	34
Table 3.2	Sign In	35
Table 3.3	Add User	36
Table 3.4	Modify User	37
Table 3.5	Sign Out	38
Table 3.6	Generate Reports	39
Table 3.7	Check Status	40
Table 3.8	Approve Application	41
Table 3.9	Reject Application	42
Table 3.10	History	43

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1	Use Case Diagram	22
Figure 3.1	Login	44
Figure 3.2	Sign up	44
Figure 3.3	Check status	45
Figure 3.4	Report generation	45
Figure 3.5	Approve application	46
Figure 3.6	Reject Application	46
Figure 3.7	Add Users	47
Figure 3.8	Modify User	47
Figure 3.9	Mark Application	48
Figure 3.10	Class Diagram	48
Figure 5.1	Intro page 2	53
Figure 5.2	Intro page 3	54
Figure 5.3	Intro page 4	54
Figure 5.4	Sign up page	55
Figure 5.5	Sign in page	55
Figure 5.6	Student Dashboard	56
Figure 5.7	New application	57
Figure 5.8	Application submission successful notification	57

Figure 5.9 Submitted application details	58
Figure 5.10 Application status	58
Figure 5.11 Profile page	59
Figure 5.12 Faculty dashboard	60
Figure 5.13 Pending applications	61
Figure 5.14 Add comment	62
Figure 5.15 Application detail page for faculty	62
Figure 5.16 Forward successful confirmation	63
Figure 5.17 Forward confirmation	63
Figure 5.18 Marking application	63
Figure 5.19 Rejected application details	64
Figure 5.20 Approved application details	64
Figure 5.21 Reports page	65
Figure 5.22 History page search bar	66
Figure 5.23 History page	66
Figure 5.24 New application uploaded notification	67
Figure 5.25 Application approved notification	67

LIST OF SYMBOLS / ABBREVIATIONS

OS	Operating System
FDD	Feature Driven Development
QD	Query Desk
AS	Authorized staff
VS	Visual Studio Code
OS	Operating System
FCM	Firebase Cloud Messaging

CHAPTER 1

INTRODUCTION

1.1 Background

Manual submissions of applications involve many steps which makes the process lengthy and time-consuming. Universities receive thousands of applications daily basis and they have to repeat the same process again and again which eventually makes the process lengthy. This requires a lot of file work and requires space for record-keeping and human resources. The involvement of authorities is necessary for the completion of the process. To avoid these major issues which are increasing day by day as the number of students is increasing rapidly. QueryDesk will help to solve his problem”.

QueryDesk is a mobile application, supported by both the most popular OS Android and IOS. The main purpose of this app is to digitalize the process of student application process in educational institutes. By using QueryDesk, a student will be able to submit an application and check the status of the application. The application category will decide which authorities to be marked. Marked authorities can check the status of the application. Admin of designated personals can generate the report of applications. From the day of admission to the end of the degree, all the applications of each student will be saved.

1.2 Problem Statements

Since the presence of personal is necessary for submission and approval of applications, can cause trouble if a student has a deadline but personals are not present on the campus. So students have to wait until the required person is available on the campus even for the submission of the application, which can ultimately cause the student to face the consequences of an exceeded deadline.

1.3 Aims and Objectives

The objectives of the thesis are shown as follows:

- i) To develop a cross-platform mobile application
- ii) To provide a digital integrated environment that will enhance the workflow
- iii) To provide one platform for different students applications
- iv) To automate the application processing flow in any organization

1.4 Scope of Project

QueryDesk is a cross-platform application that will provide a platform where the student can upload their applications and authority can do their work digitally which will minimize their time-consumption and workload. In the manual process, if a student's application gets rejected, he or she will not get notified until he or she calls or pay a visit to the respective office. With the help QueryDesk application, we will provide push notification if our application gets rejected or if not then at which phase the application is. In This COVID-19 pandemic, the work from home is in trend if any

of the authorities are working from home, they can carry out their respective tasks easily.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 User Classes and Characteristics

This software is created for two main user classes:

- i) Students (Knowledgeable users)
- ii) Authorized staff (Knowledgeable users)
- iii) Administrator (Expert user)

Authorized staff – Authorized staff will be able to perform the following activities:

- Sign-in
- Mark application
- Comments
- Reject application
- Sign-out
- History
- Reports

Students – Any student will be able to perform the following activities:

- Sign-up
- Sign-in
- Submit a new application
- Check the status of an application
- Sign-out
- Can view the list of applications submitted by students

Administrator – An admin will be able to perform the following activities:

- Sign in
- Generate history
- Manage other users
- Sign out
- Reports
- Create users

2.2 Operating Environment

Mobile device with the following (minimum) specifications:

- 2 GB RAM
- 1 GB free storage
- Camera

- Android 6.0 marshmallow
- IOS 9.0

2.3 Development System

- Android Studio
- Visual Studio Code
- Flutter [1]
- Adobe XD
- Firebase

2.3.1 Design and Implementation Constraints

QueryDesk app is a mobile application, and the front end of the Application is designed using Abode XD and Flutter. For the online database, Firebase [2] is used. It will be developed using FDD [3] which divides complex structures into subparts. This model recurrence last two phases i.e., Design by feature and Develop by Feature until all features will complete. The internet connection is also a constraint for the application. Since the application fetches data from firebase, there must be an internet connection for the application to function

2.4 Assumptions and Dependencies

We are assuming that the user already has the knowledge required how to use a smartphone. Internet connection is a must dependency for this software. Without these services, QueryDesk won't be able to run.

2.5 External Interface Requirements

Following are the external interface requirements of this project:

2.5.1 User Interfaces

- Full-screen application
- Splash screen
- UI/UX includes the following components:
 - Buttons
 - TextViews
 - Transitions
 - Google defined UI constraint standards
 - Material design theme

2.6 Software Interfaces

Following are the required software interfaces:

Table 2.1 Software Interfaces

Android Version	6.0+	This app will run on mobiles that are running Android version 6.0+
IOS	9.0+	This app will run on mobiles that are running IOS version 9.0+
Database	FireBase	Firestore will be used to save the record of the users.
Tools	Android Studio	Android Studio and Visual Studio Code is the development platform that'll be used to develop this application using the Flutter framework.

2.7 System Use Cases

System use case diagram of this project shows in figure 2.1:

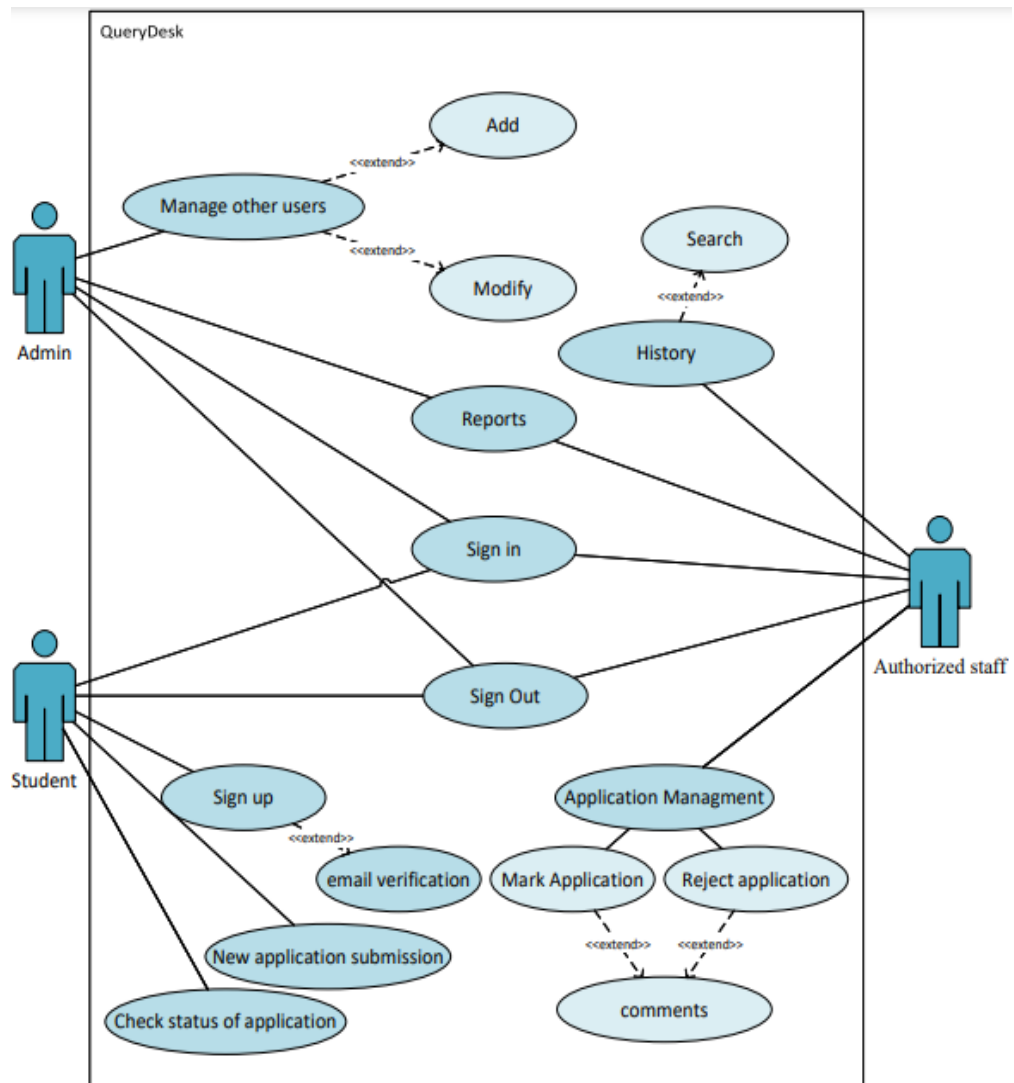


Figure 2.1 Use Case Diagram

2.7.1 Sign-Up as a Student (U1)

Table 2.2 Sign-Up as a Student

	Name	Sign-Up as a Student
1.	Use-Case ID	U1
2.	Objective	Students will Sign-Up with valid credentials.
3.	Priority	Normal
4.	Actors	Student
5.	Flow of Events	<ul style="list-style-type: none"> • Open App • Click on Sign-Up Button • Enter Required Credentials • Press Sign-up button • Verification of credentials
5.1	Basic Flow	Same as Flow of Events.
5.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If the user already exists in the database. The user will be asked to log in if his record already exists • If all required fields are not filled. An error message is displayed to fill all the fields.
5.3	Exception Flow(s)	<ul style="list-style-type: none"> • Invalid Username • Invalid Password
6.	Includes	No other use case
7.	Preconditions	The user must be not already registered.
8.	Postconditions	The user gets registered.
9.	Notes/Issues	None.

2.7.2 Sign-In as Student (U2)

Table 2.3 Sign-In as a Student

	Name	Sign-In as a Student
1.	Use-Case ID	U2
2.	Objective	Students will Sign-In with valid credentials.
3.	Priority	Normal
4.	Actors	Student
5.	Flow of Events	<ul style="list-style-type: none"> • Open App • Enter Required Credentials • Press Sign-In button
5.1	Basic Flow	Same as Flow of Events.
5.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If the User's record does not exist in the database. The user will be asked to Sign-up first. • If wrong credentials are entered during sign in it will ask to sign in with the right credentials. • If the user enters the wrong password press forgot password • Password reset link via email
5.3	Exception Flow(s)	If all required fields are not filled error message is displayed to fill all the fields.
6.	Preconditions	The user must be already registered.
7.	Postconditions	The user gets registered.
8.	Notes/Issues	None.

2.7.3 Sign-In as Faculty Member (U3)

Table 2.4 Sign-In as a Faculty Member

	Name	Sign-In as a Faculty Member
1.	Use-Case ID	U3
2.	Objective	Faculty Member will Sign-In with valid credentials.
3.	Priority	High
4.	Source	Faculty
5.	Actors	Faculty Member
6.	Flow of Events	<ul style="list-style-type: none"> • Open App • Enter Required Credentials • Press Sign-In button
6.1	Basic Flow	Same as Flow of Events.
6.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If the User's record does not exist in the database. The user will be asked to Sign-up first. • If wrong credentials are entered during sign in it will ask to sign in with the right credentials. • If the user enters the wrong password press forgot password • Password reset link via email
6.3	Exception Flow(s)	If all required fields are not filled error message is displayed to fill all the fields.
7.	Preconditions	The user must be already registered.
8.	Postconditions	The user gets registered.
9.	Notes/Issues	None.

2.7.4 Sign-In as Administrator (U4)

Table 2.5 Sign-In as Administrator

	Name	Sign-In as Administrator
1.	Use-Case ID	U4
2.	Objective	The administrator will Sign-In with valid credentials.
3.	Priority	High
4.	Source	Administrator
5.	Actors	Administrator
6.	Flow of Events	<ul style="list-style-type: none"> • Open App • Enter Required Credentials • Press Sign-In button
6.1	Basic Flow	Same as Flow of Events.
6.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If wrong credentials are entered during sign in it will ask to sign in with the right credentials. • If the user enters the wrong password press forgot password • Password reset link via email
6.3	Exception Flow(s)	If all required fields are not filled error message is displayed to fill all the fields.
8.	Preconditions	Must have valid credentials
9.	Postconditions	None.
10.	Notes/Issues	None.

2.7.5 New Application Submission (U5)

Table 2.6 New Application Submission

	Name	New Application Submission
1.	Use-Case ID	U5
2.	Objective	Students can submit a new application.
3.	Priority	High
4.	Actors	Student
5.	Flow of Events	<ul style="list-style-type: none"> • Open App • Enter Required Credentials • Press Sign-In button • Click on new application • Fill in the required information • Submit application
5.1	Basic Flow	Same as Flow of Events.
5.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If the User's record does not exist in the database. The user will be asked to Sign-up first. • If wrong credentials are entered during sign in it will ask to sign in with the right credentials. • If the required template isn't on the list, it will show items not available.
5.3	Exception Flow(s)	<ul style="list-style-type: none"> • If all required fields are not filled error message is displayed to fill all the fields. • If the user enters the wrong password press forgot password • Password reset link via email
6.	Preconditions	The user must be Signed-In.
7.	Postconditions	The user submits an application.
8.	Notes/Issues	None.

2.7.6 Approve Application (U6)

Table 2.7 Accept Application

	Name	Approve Application
1.	Use-Case ID	U6
2.	Objective	A authorized staff can accept the application
3.	Priority	High
4.	Source	Authorized staff.
5.	Actors	Authorized staff
6.	Flow of Events	<ul style="list-style-type: none"> • Open App • Login • View the list of submitted applications. • Accept application.
6.1	Basic Flow	Same as Flow of Events.
6.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If the User's record does not exist in the database. The user will be asked to Sign-up first. • If wrong credentials are entered during sign in it will ask to sign in with the right credentials.
6.3	Exception Flow(s)	<ul style="list-style-type: none"> • If all required fields are not filled error message is displayed to fill all the fields. • If the user enters the wrong password press forgot password • Password reset link via email
7.	Preconditions	The user must be Signed-In.
8.	Postconditions	User-submitted application against certain issues has been accepted.
9.	Notes/Issues	None.

2.7.7 Reject Application (U7)

Table 2.8 Reject Application

	Name	Reject Application
1.	Use-Case ID	U7
2.	Objective	A faculty member can Reject the application
3.	Priority	High
4.	Source	Authorized staff.
5.	Actors	Authorized staff
6.	Flow of Events	<ul style="list-style-type: none"> • Open App • Login • View the list of submitted applications. • Select one • Reject application.
6.1	Basic Flow	Same as Flow of Events.
6.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If the User's record does not exist in the database. The user will be asked to Sign-up first. • If wrong credentials are entered during sign in it will ask to sign in with the right credentials.
6.3	Exception Flow(s)	<ul style="list-style-type: none"> • If all required fields are not filled error message is displayed to fill all the fields. • If the user enters the wrong password press forgot password
7.	Preconditions	The user must be Signed-In.
8.	Postconditions	User-submitted application against certain issue has been Rejected.
9.	Notes/Issues	None.

2.7.8 Application Status (U8)

Table 2.9 Application Status

	Name	Application Status
1.	Use-Case ID	U8
2.	Objective	Students can view the current status of their applications.
3.	Priority	Normal
4.	Actors	Student
5.	Flow of Events	<ul style="list-style-type: none"> • Open App • Login • View Status
5.1	Basic Flow	Same as Flow of Events.
5.2	Alternate Flow(s)	<ul style="list-style-type: none"> • If the User's record does not exist in the database. The user will be asked to Sign-up first. • If wrong credentials are entered during sign in it will ask to sign in with the right credentials. • If the required application template isn't on the list it will show an item not available.
5.3	Exception Flow(s)	<ul style="list-style-type: none"> • If all required fields are not filled error message is displayed to fill all the fields. • If the user enters the wrong password press forgot password • Password reset link via email
6.	Preconditions	The application must be submitted
7.	Postconditions	Users have viewed the status of their applications.
8.	Notes/Issues	None.

2.8 Other Non-functional Requirements

Non-functional requirements of this project are:

2.8.1 Performance Requirements

A slow internet connection may impact the performance of the application. A good and stable internet connection is required.

2.8.2 Safety Requirements

To provide the users with the best application experience we will update our application from time to time to fix the bugs and errors.

2.8.3 Security Requirements

The database has to be reached securely and its data should not be broken. It also should not change except for integrant updates. Moreover, since our dataset contains some personal information of the user such as user id, applications he/she submitted, security design is important in the application service.

2.8.4 Software Quality Attributes

Software quality attributes of this project are:

- **Availability:** The application will be available for the user 24/7.
- **Flexibility:** The application would be flexible for any type of user.
- **Usability:** The application should be user-friendly for the user. The users easily understand how to use the application.
- **Testability:** The application should be easy to test at each level and find the bugs/defect at each level of development and remove the defects easily.

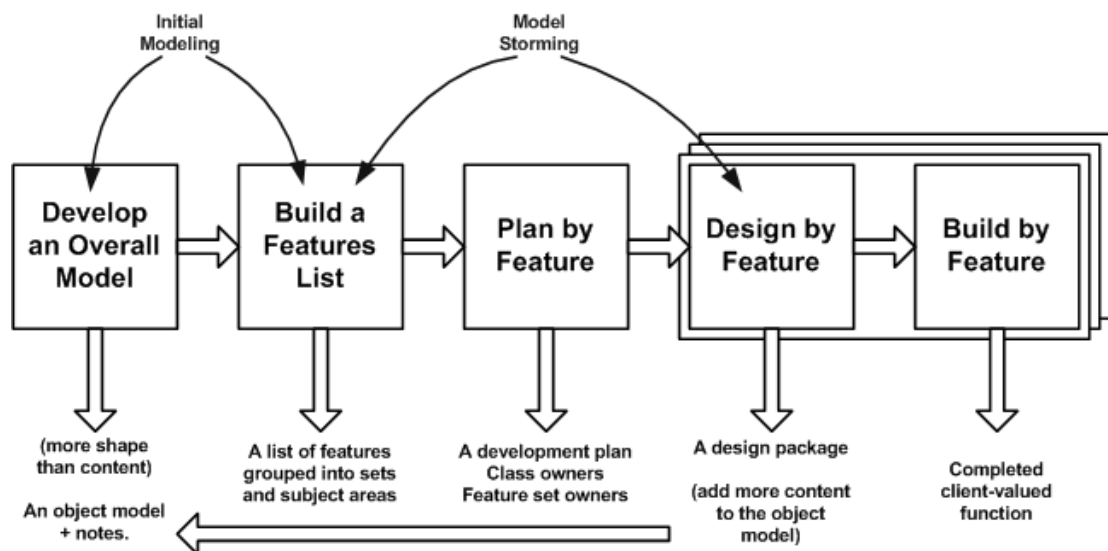
- **Reusability:** The application is divided into different modules of coding. These modules can be used across the application.
- **Maintainability:** The design of this application is understandable and can be enhanced according to the business needs

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Features-Driven development

FDD is an incremental and iterative software development methodology. It is an Agile method for developing software. FDD combines a variety of well-known industry best practices into a unified totality. These practices are driven from a client-valued functionality perspective



Copyright 2002-2005 Scott W. Ambler
Original Copyright S. R. Palmer & J.M. Felsing

3.2 Use-case Description

3.2.1 Sign up

Table 3.1 Sign up

Use Case ID:	QD-I		
Use Case Name:	Sign-up		
Used By:	Student		
Date Created:	08-09-2021	Date Last Updated:	25-12-2021
Actor:	Student,		
Priority:	High		
Preconditions:	<p>The user is required to enter the information that is necessary to become a member of the system.</p> <p>The user's identity has been authenticated.</p> <p>The user's mobile has sufficient free memory available to launch the app.</p>		
Postconditions:	<p>After the user Signup on the application</p> <p>The number of users on the application has been updated</p>		
Normal Course of Events:	<p>User get verification notification</p> <p>Enters valid information following regular expression</p>		
Alternative Courses:			
Exceptions:	<p>Internet shutdown</p> <p>Error 404</p>		
Includes:	<p>To sign up on the application, the user is required to visit the application first but this use case is not preceded by any further use case</p>		
Notes & Issues	None		

3.2.2 Sign In

Table 3.2 Sign In

Use Case ID:	QD-II		
Use Case Name:	Sign In		
Used By:	Student, Authorized staff, Admin		
Date Created:	08-09-2021	Date Last Updated:	25-12-2021
Actor:	Student, Authorized staff, Admin		
Priority:	High		
Preconditions:	The user is required to have a Password and ID		
Postconditions:	After the user signup on the application <ul style="list-style-type: none"> The number of users on the application has been updated 		
Normal Course of Events:	User successfully sign-in on the page using various methods and become able to access various things related to his profile In this particular case <ol style="list-style-type: none"> Authorized staff can perform application management Students can check the status of an application and submit the application Admin can manage the both above and other various tasks 		
Alternative Courses:	None		
Exceptions:	<ul style="list-style-type: none"> Internet shutdown Error 404 		
Includes:	In this use case, the user must sign up on the application first.		
Notes & Issues:	None		

3.2.3 Add Users

Table 3.3 Add User

Use Case ID:	QD-III		
Use Case Name:	Add user		
Used By:	Admin		
Date Created:	08-09-2021	Date Last Updated:	25-12-2021
Actor:	Admin		
Priority:	Average		
Preconditions:	The user is required to have valid credentials. After the add user on the application number of users on the application has been updated		
Postconditions:	<ul style="list-style-type: none"> User must be categorized i.e., Authorized staff or Student 		
Normal Course of Events:	<ol style="list-style-type: none"> Authorized staff can sign in/out and perform the application management task Student can perform their available task 		
Alternative Courses:	<ul style="list-style-type: none"> As of step 1 user must get notified 		
Exceptions:	<ul style="list-style-type: none"> Internet shutdown Error 404 		
Includes:	User should write appropriate ID, name, Email ID		
Notes & Issues:	None		

3.2.4 Modify Users

Table 3.4 Modify User

Use Case ID:	QD-IV		
Use Case Name:	Modify user		
Used By:	Admin		
Date Created:	08-09-2021	Date Last Updated:	25-12-2021
Actor:	Admin		
Priority:	High		
Preconditions:	<p>The user must be an already existing user.</p> <p>The user is required to have a Validated Password and ID.</p> <p>After the modified user on the application number of users on the application remains the same.</p> <p>User must be admin</p>		
Postconditions:	<ul style="list-style-type: none"> User must Be categorized i.e., Authorized staff or Student 		
Normal Course of Events:	<ol style="list-style-type: none"> Form the pre-existing user admin can modify them e.g., for Authorized staff his /her roles may get change or a new person take the charges of that designation Student name of other information may need correction 		
Alternative Courses:	<ul style="list-style-type: none"> As of step 1, the admin may ask user Authorized staff certain questions to make user profile properly as user login into the system. 		
Exceptions:	<ul style="list-style-type: none"> Internet shutdown Error 404 		
Notes & Issues:	None		

3.2.5 Sign Out

Table 3.5 Sign Out

Use Case ID:	QD-V		
Use Case Name:	Sign Out		
Used By:	Student, Authorized staff, Admin		
Date Created:	08-09-2021	Date Last Updated:	20-12-2021
Actor:	Student, Authorized staff, Admin		
Priority:	High		
Preconditions:	The user must have valid login credentials. The user must be signed in.		
Postconditions:	User must be categorized according to application design		
Normal Course of Events:	After the user click on sign out user will safely sign out the application immediately		
Alternative Courses:	As of step 1 system may not respond immediately due to network issues		
Exceptions:	Internet shutdown Error 404		

3.2.6 Generate Reports

Table 3.6 Generate Reports

Use Case ID:	QD-VI		
Use Case Name:	Generate reports		
Used By:	Admin, Authorized staff		
Date Created:	08-09-2021	Date Last Updated:	25-12-2021
Actor:	Admin, Authorized staff		
Priority:	High		
Preconditions:	Users must have admin rights. The user is required to have a Validated Password and ID.		
Normal Course of Events:	<ul style="list-style-type: none"> Admin/Authorized staff sign-in and generate reports of all applications, including rejected applications. 		
Alternative Courses:	<ul style="list-style-type: none"> Reports generation may be time taking. 		
Exceptions:	<ul style="list-style-type: none"> Delete application by admin is not involved 		
Notes & Issues	None		

3.2.7 Check Status

Table 3.7 Check Status

Use Case ID:	QD-VIII		
Use Case Name:	Check Status		
Used By:	Student		
Date Created:	08-09-2021	Date Last Updated:	25-12-2021
Actor:	Student		
Priority:	High		
Preconditions:	<p>The user must be a student.</p> <p>The user must be logged in.</p> <p>An application must be submitted to check its status.</p>		
Postconditions:	The user is a student and applications are submitted.		
Normal Course of Events:	<p>Sign in with valid credentials</p> <p>Click on status</p> <p>Click on the application submitted by the student</p>		
Alternative Courses:	<ul style="list-style-type: none"> • No application was found. • Failed to get status due to an exception. 		
Exceptions:	<ul style="list-style-type: none"> • Internet shutdown • Error 404 		

3.2.8 Approve application

Table 3.8 Approve Application

Use Case ID:	QD-IX		
Use Case Name:	Approve application		
Used By:	Authorized staff		
Date Created:	08-09-2021	Date Last Updated:	20-12-2021
Actor:	Authorized staff		
Priority:	High		
Preconditions:	User must be Authorized staff. The user must be logged in. Applications must be submitted by the applicant.		
Postconditions:	Users have Authorized staff rights and applications must be pending.		
Normal Course of Events:	Authorized staff member login and approve the application submitted by the student and move to next phase.		
Alternative Courses:	<ul style="list-style-type: none"> • No application was found. • Failed to approve due to an exception. 		
Exceptions:	<ul style="list-style-type: none"> • Internet shutdown • Error 404 		

3.2.9 Reject application

Table 3.9 Reject Application

Use Case ID:	QD-X		
Use Case Name:	Reject application		
Used By:	Authorized staff		
Date Created:	08-09-2021	Date Last Updated:	08-09-2021
Actor:	Authorized staff		
Priority:	High		
Preconditions:	User must be Authorized staff. The user must be logged in. The application is submitted by the student.		
Postconditions:	Users have Authorized staff rights and applications are submitted. Authorized staff must add comments		
Normal Course of Events:	An authorized staff member logs in and rejects the application submitted by the student.		
Alternative Courses:	<ul style="list-style-type: none"> • No application was found. • Failed to reject due to an exception. 		
Exceptions:	<ul style="list-style-type: none"> • Internet shutdown • Error 404 		

3.2.10 History

Table 3.10History

Use Case ID:	QD-XI		
Use Case Name:	History		
Used By:	Admin, Authorized staff		
Date Created:	08-09-2021	Date Last Updated:	25-12-2021
Actor:	Admin, Authorized staff		
Priority:	High		
Preconditions:	<p>Users must have admin rights.</p> <p>The user is required to have a Validated Password and ID.</p> <p>An application must be submitted</p>		
Normal Course of Events:	<ul style="list-style-type: none"> • Admin/Authorized staff sign in and see all the applications submitted yet and on which level they are now. • The search of a specific application by enrollment 		
Alternative Courses:	<ul style="list-style-type: none"> • History generation may be time taking. 		
Exceptions:	<ul style="list-style-type: none"> • Internet shutdown • Error 404 		

3.3 Sequence Diagrams

3.3.1 Sign in

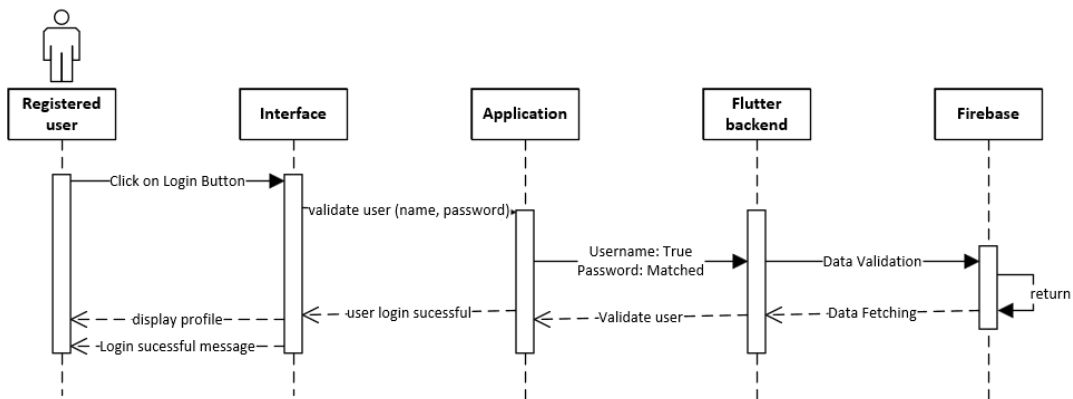


Figure 3.1 Login

3.3.2 Sign up

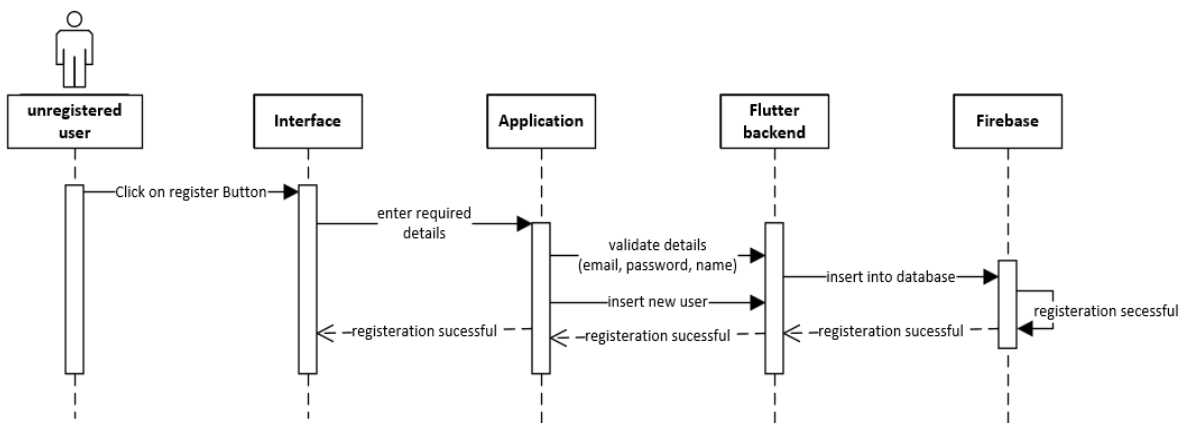


Figure 3.2 Sign up

3.3.3 Check status

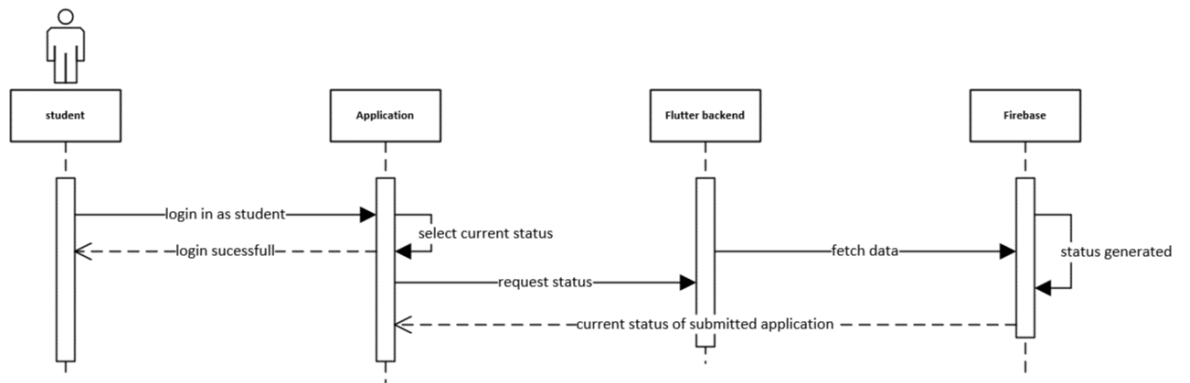


Figure 3.3 Check status

3.3.4 Report generation

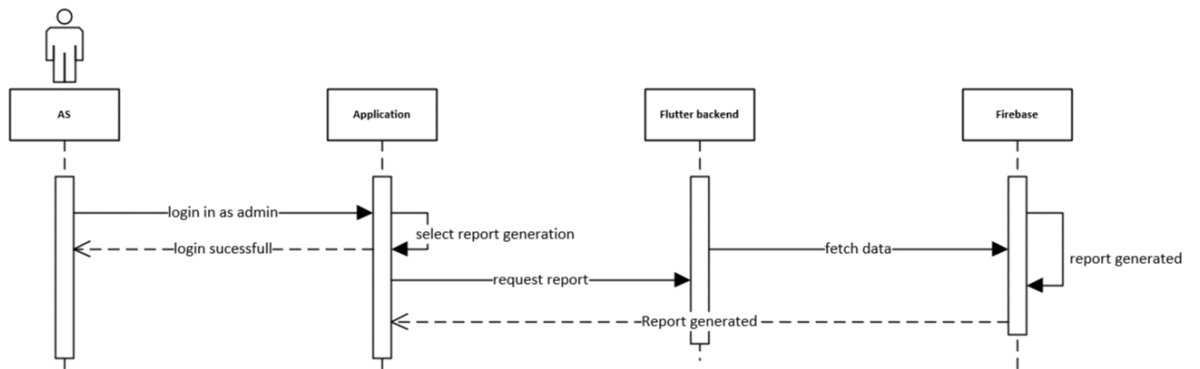


Figure 3.4 Report generation

3.3.5 Approve application

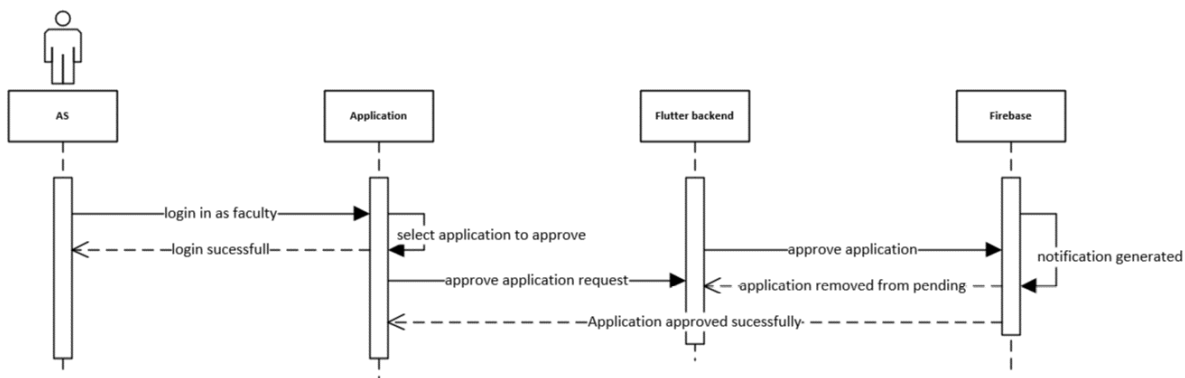


Figure 3.5 Approve application

3.3.6 Reject application

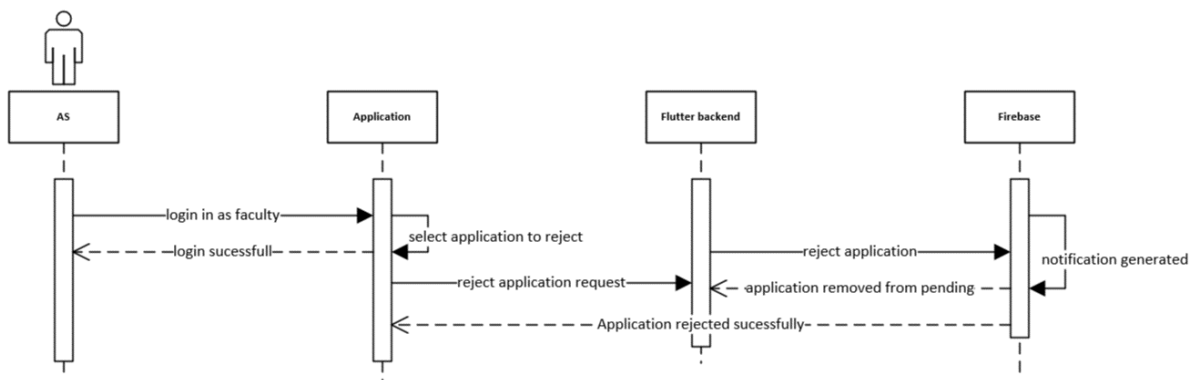


Figure 3.6 Reject Application

3.3.7 Add user

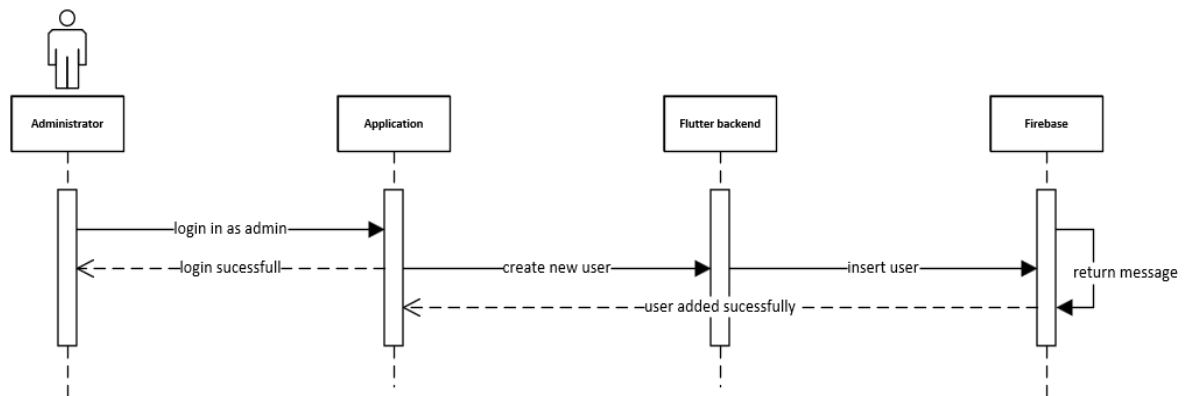


Figure 3.7 Add Users

3.3.8 Modify user

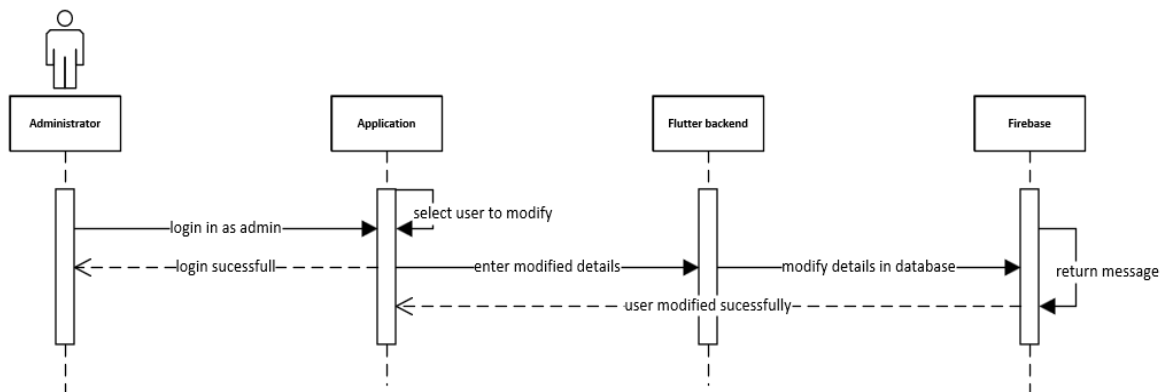


Figure 3.8 Modify User

3.3.9 Mark Application

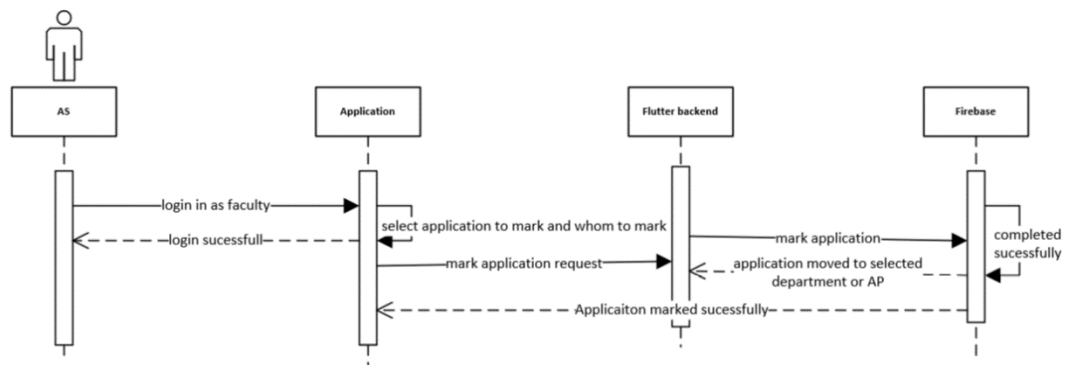


Figure 3.9 Mark Application

3.4 Class Diagram

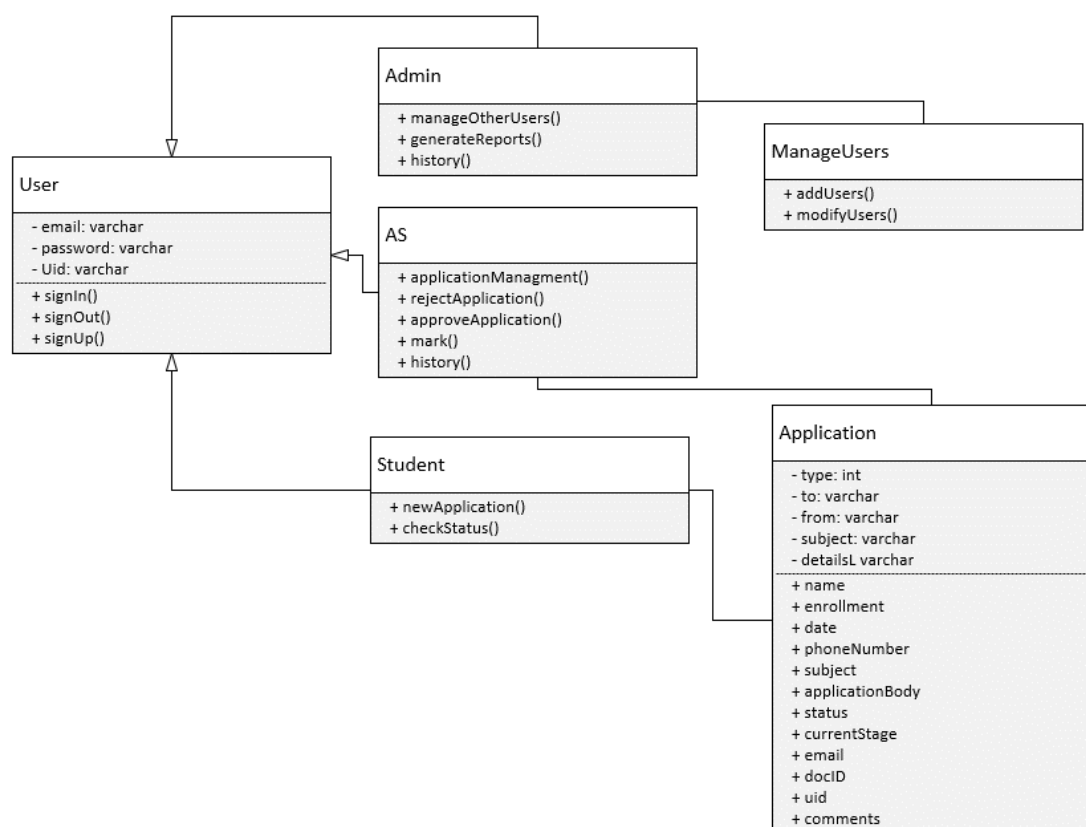


Figure 3.10 Class Diagram

3.5 ERD Diagram

The Firebase databases, either real-time databases or the cloud firestore, are document-oriented NoSQL databases. There are indeed no tables, just collections of documents, where each document is a JSON-like structure, and documents could contain other embedded collections or documents.[4] As said before firebase uses a NoSQL database. It uses JSON objects to be precise.

Consider it like this.

You have three employees then the JSON tree will be formed like this. [5]

- Employees
 - Employee 1
 - Name
 - Address
 - Employee 2
 - Name
 - Address

So this whole set is considered as one JSON object. Whenever you try to retrieve anything then you have to capture the entire object and traverse through it.

CHAPTER 4

IMPLEMENTATION

4.1 Languages used for implementation

4.1.1 Dart

Dart is an Open-Source, Client-Optimized programming language used for creating high-performance and fast applications. It is used to create a mobile application, write software for the desktop, or design a site. It's an interpretable language supported by Google alongside numerous features which made it popular among programmers. Dart is also known as an alternative to JavaScript. The use of it is increased when Google introduced Flutter intending to add mobile programming capabilities

4.2 Framework:

4.2.1 Flutter

Flutter is an Open-source UI, cross-platform framework for developing a native mobile application. Flutter allows us to build a mobile application for Android and IOS with a single code-based and programming language called Dart. The flutter framework consists of both a software development kit (SDK) and their widget-based UI library.it contains a variety of reusable UI elements.

4.3 Tools

4.3.1 VS code

Microsoft Visual Studio Code is a standalone source code editor that runs on multiple operating systems. It is used to develop computer programs, as well as websites, web apps, web services, and mobile apps. It supports many programming languages and a set of features that differs per language. VS Code provides initial support including syntax highlighting etc.

4.3.2 Android Studio

Android studio is an official integrated development environment for the development of an android based application. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

4.3.3 Android Emulator

The android emulator provides simulations of android devices on your computer which helps us to test our application on a variety of devices without needing to have each physical device. It came along with a predefined configuration for multiple android based Phones, Tablets. Testing in an emulator is way faster than physical devices.

4.3.4 Abode XD

Abode XD is a prototyping tool for user experiences and interaction designers. Adobe XD features are useful for creating wireframes prototypes UI designing for digital products such as mobile apps and websites.

4.4 Online services

4.4.1 Firebase

Firebase is a backend as a service. It provides developers a variety of services along with tools to help them to develop a quality application that helps them to grow their user base, and earn. It is built on googles infrastructure which provides a variety of storage services

4.4.2 Cloud Firestore

Cloud firestore is a scalable database for mobile, web, server development from firebase and google cloud.it tends to keep your data in sync across client apps through realtime listeners and offers offline support for web and mobile applications so that we can develop an app without worrying about network latency or internet connectivity.

4.4.3 Firebase Cloud Messaging

FCM is a cross-platform solution that lets you safely send a message to multiple or single devices over the internet. In our case, users will get notifications according to their access levels.

CHAPTER 5

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 Introduction

This is the introduction of the application which tells the main features of QueryDesk and this will show only at once when the user downloads the application and runs it for the first time.

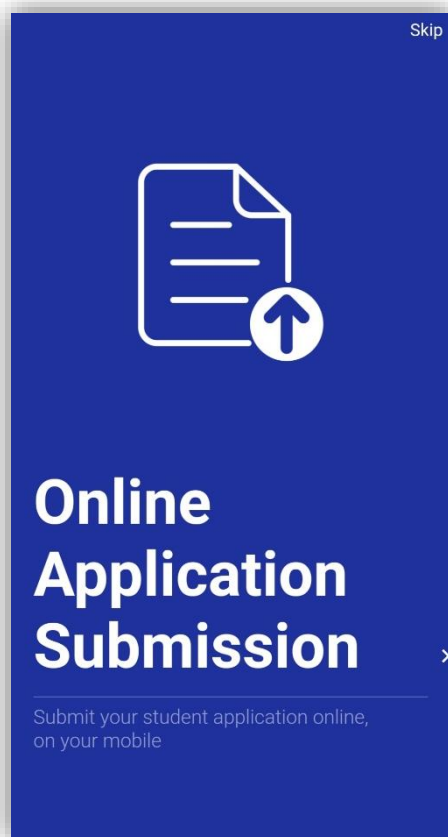


Figure 5. 5.1 Intro Page 1

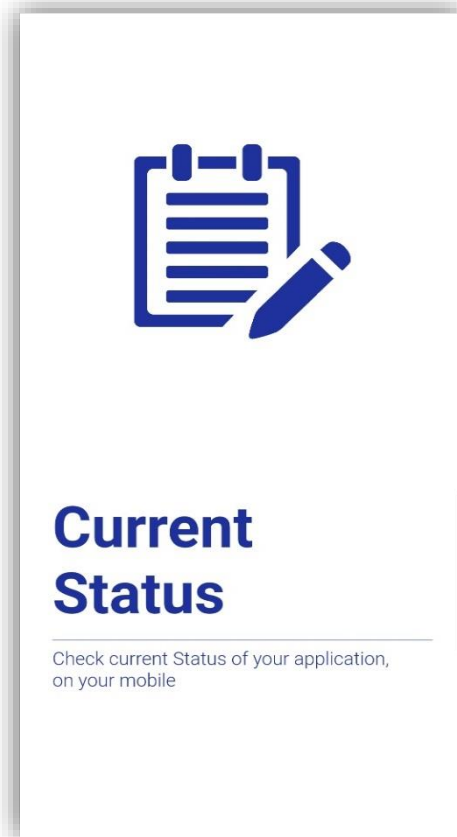


Figure 5.1 Intro page 2

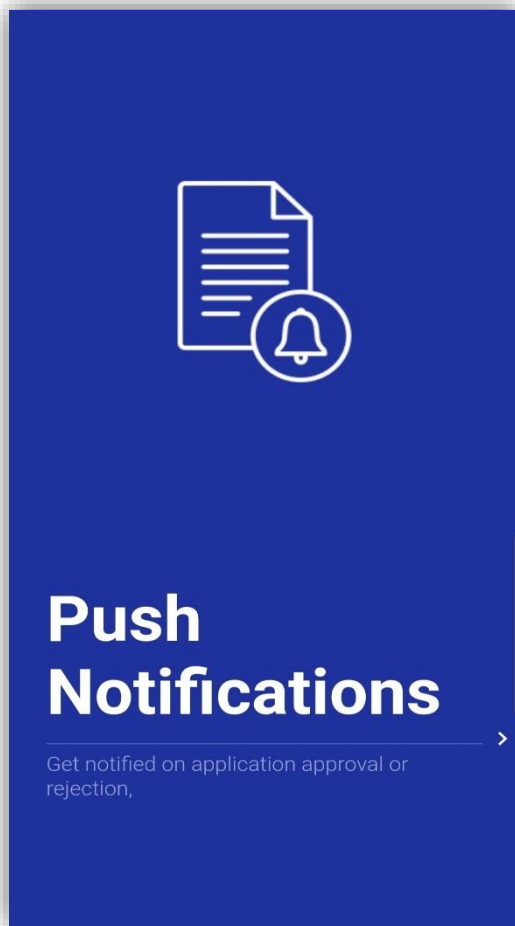


Figure 5.2 Intro page 3

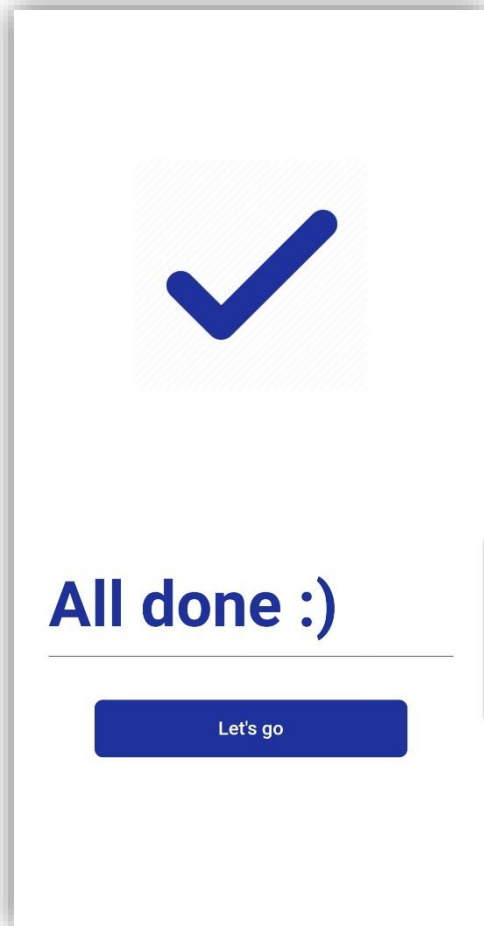
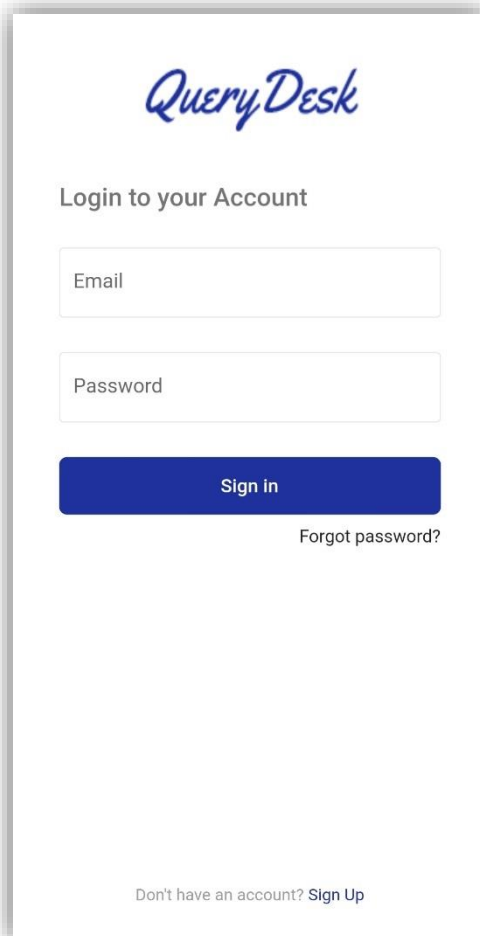


Figure 5.3 Intro page 4

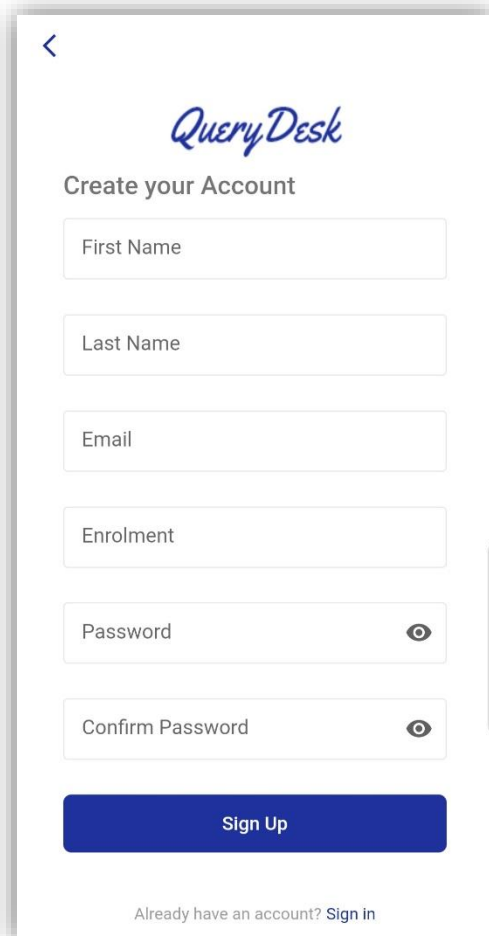
5.2 Sign In and Sing Up

The Sign In and Sign Up is simple as other mobile applications had with the additional features of Email verification which will support the confidentiality triad.



The screenshot shows the 'Sign In' page for 'QueryDesk'. At the top is the 'QueryDesk' logo in a blue script font. Below it is the heading 'Login to your Account'. There are two input fields: 'Email' and 'Password'. A blue button labeled 'Sign in' is positioned below the password field. To the right of the button is a link for 'Forgot password?'. At the bottom, there is a link that says 'Don't have an account? Sign Up'.

Figure 5.5 Sign in page



The screenshot shows the 'Sign Up' page for 'QueryDesk'. At the top is the 'QueryDesk' logo in a blue script font. Below it is the heading 'Create your Account'. There are five input fields: 'First Name', 'Last Name', 'Email', 'Enrolment', and 'Password'. The 'Password' field has an eye icon to its right. Below the 'Password' field is a 'Confirm Password' field, also with an eye icon to its right. A blue button labeled 'Sign Up' is positioned below the 'Confirm Password' field. At the bottom, there is a link that says 'Already have an account? Sign in'.

Figure 5.4 Sign up page

5.3 Student Dashboard

After Sign Student will be directed towards this screen which has the following feature:

- New Application tab
- Status Tab
- Profile Tab

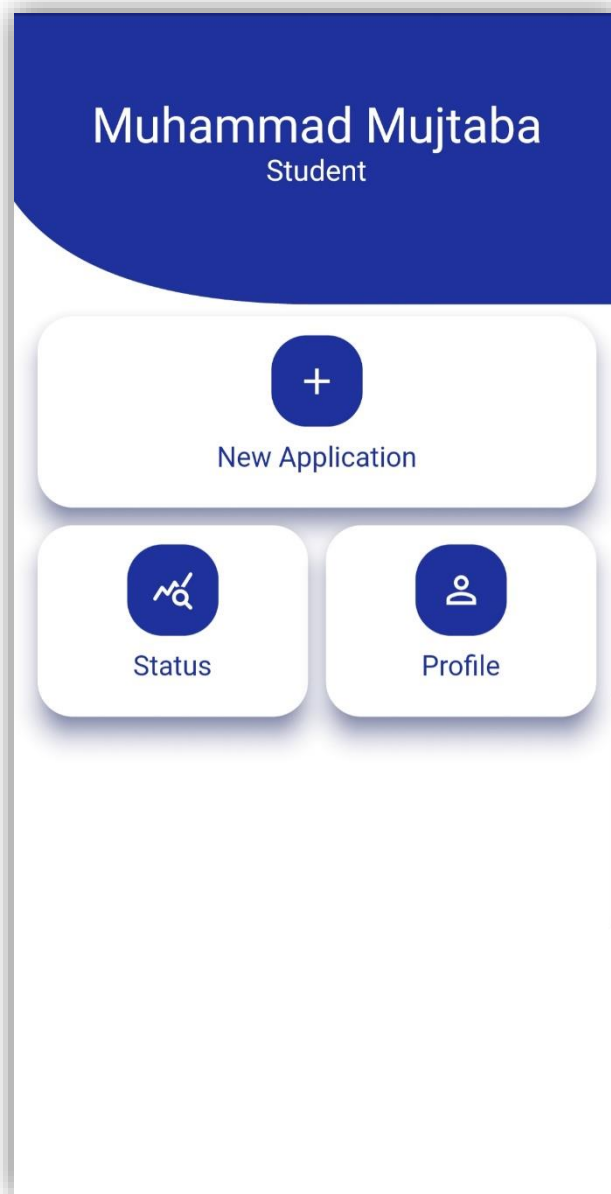
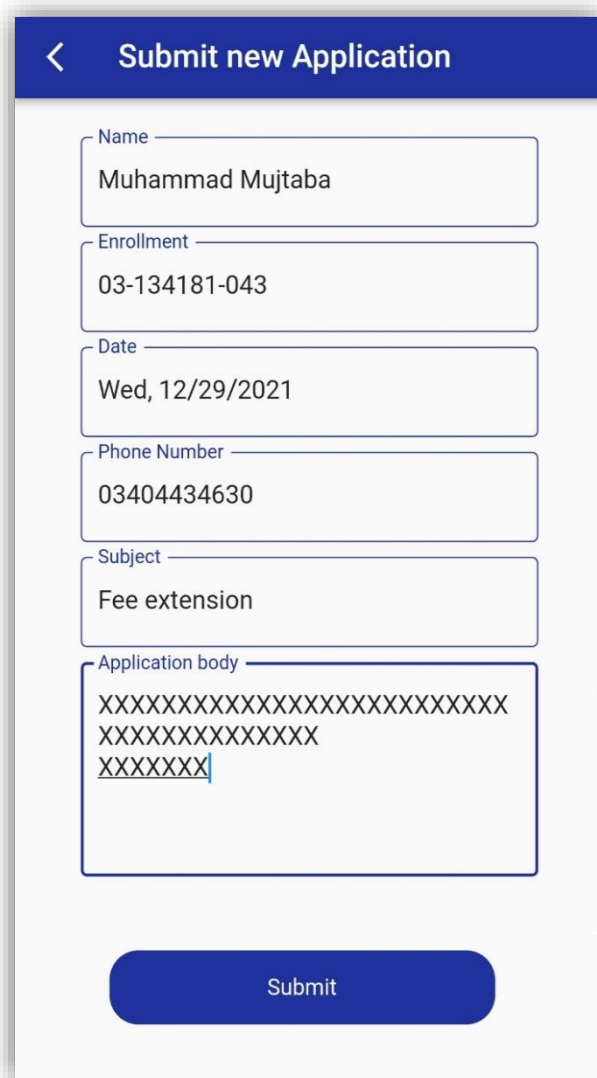


Figure 5.6 Student Dashboard

5.4 New Application Submission

Students can select the subject from a predefined list of subjects or add a custom subject. After submitting the application receive the notification as attached in the following image and re-directed to the main page



The screenshot shows a mobile application interface for submitting a new application. The title bar is blue with a white back arrow and the text "Submit new Application". Below the title bar are several input fields: "Name" with the value "Muhammad Mujtaba", "Enrollment" with "03-134181-043", "Date" with "Wed, 12/29/2021", "Phone Number" with "03404434630", "Subject" with "Fee extension", and "Application body" with placeholder text "XXXXXXXXXXXXXXXXXXXXXXXXXXXX", "XXXXXXXXXXXXXXXXXXXX", and "XXXXXXX|". A blue "Submit" button is located at the bottom of the form.

Figure 5.7 New application

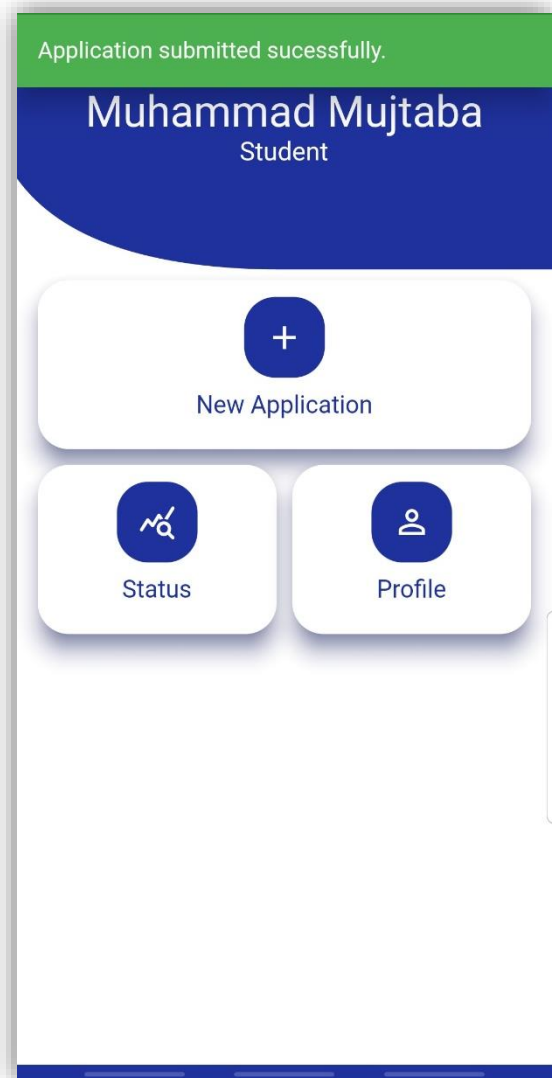


Figure 5.8 Application submission successful notification

5.5 Application Status

After clicking on the status tab student can see the list of his/her application submissions. You can view the current stage of your application i.e. approved, rejected, or pending.

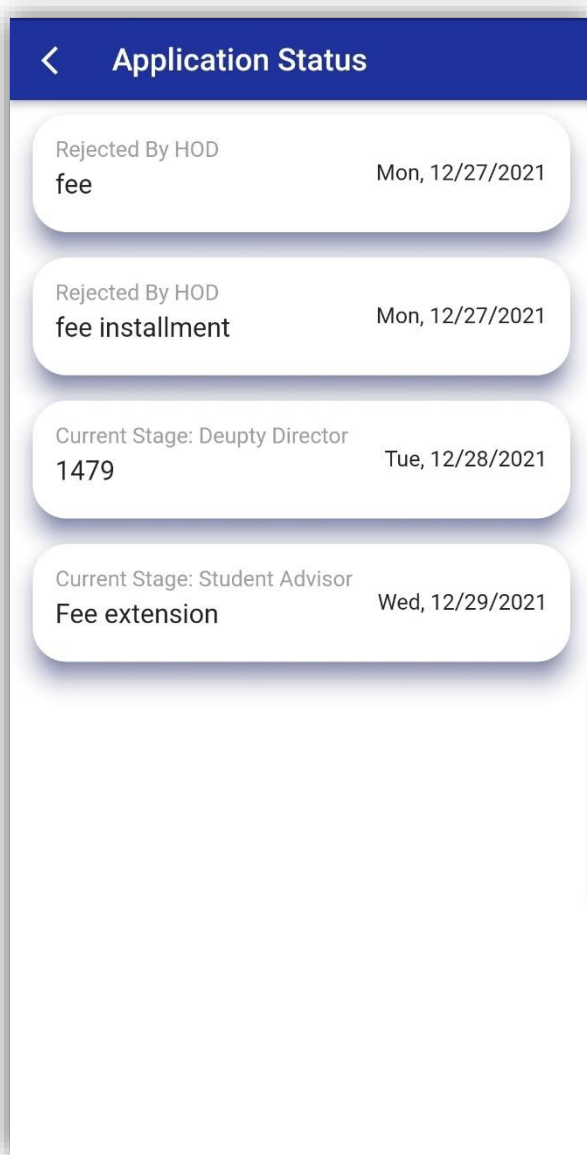


Figure 5.10 Application status

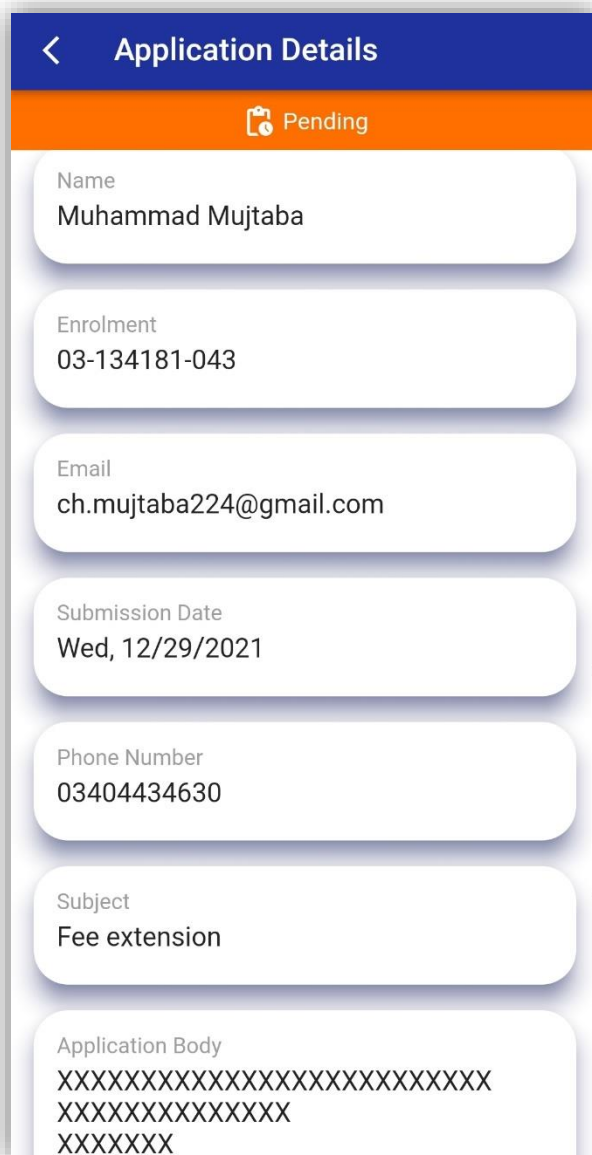


Figure 5.9 Submitted application details

5.7 Profile Tab

- Details of the user
- Sign out button
- Access level information

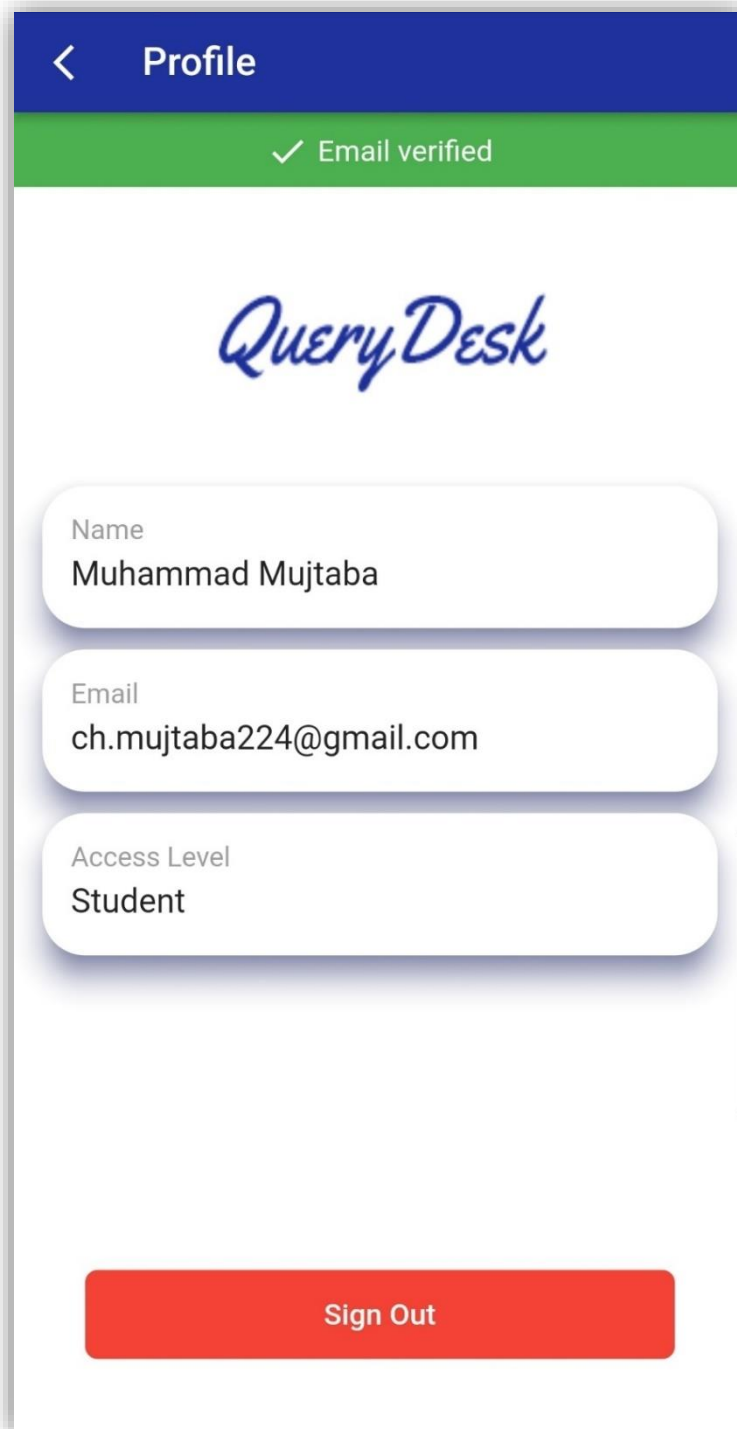


Figure 5.11 Profile page

5.8 Authorized Dashboard

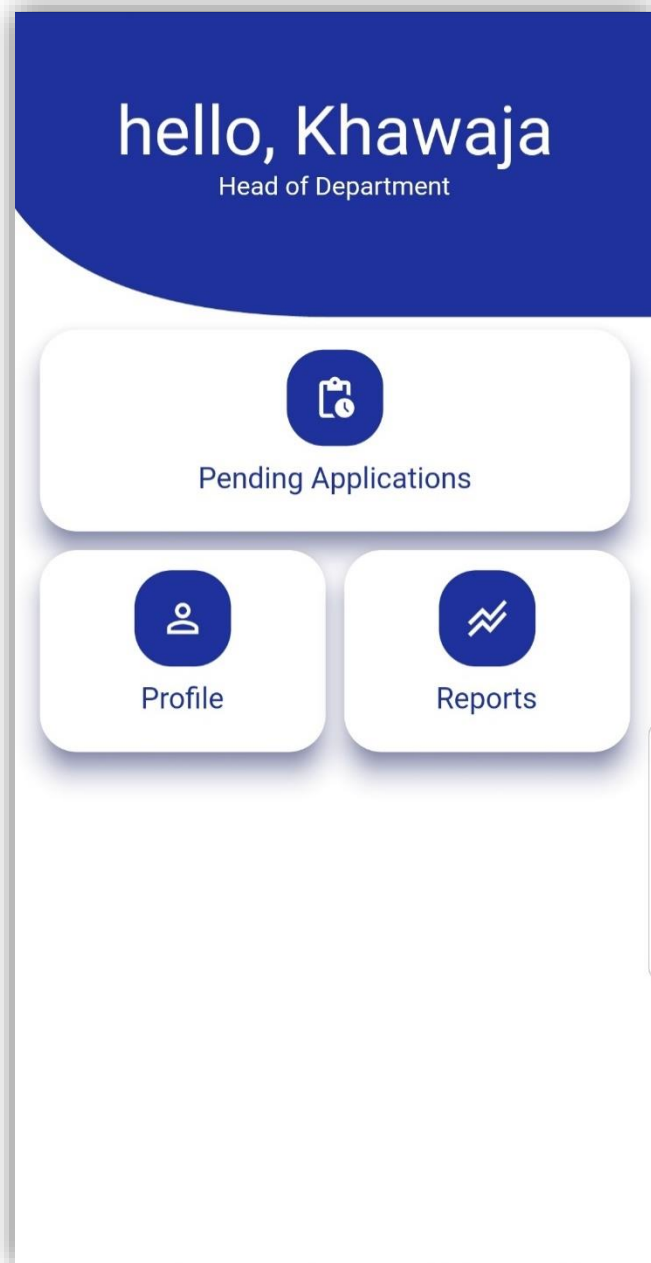


Figure 5.12 Faculty dashboard

5.9 Pending Application

A list of all submitted applications is shown here. Authorized staff can view the application details by clicking on them.

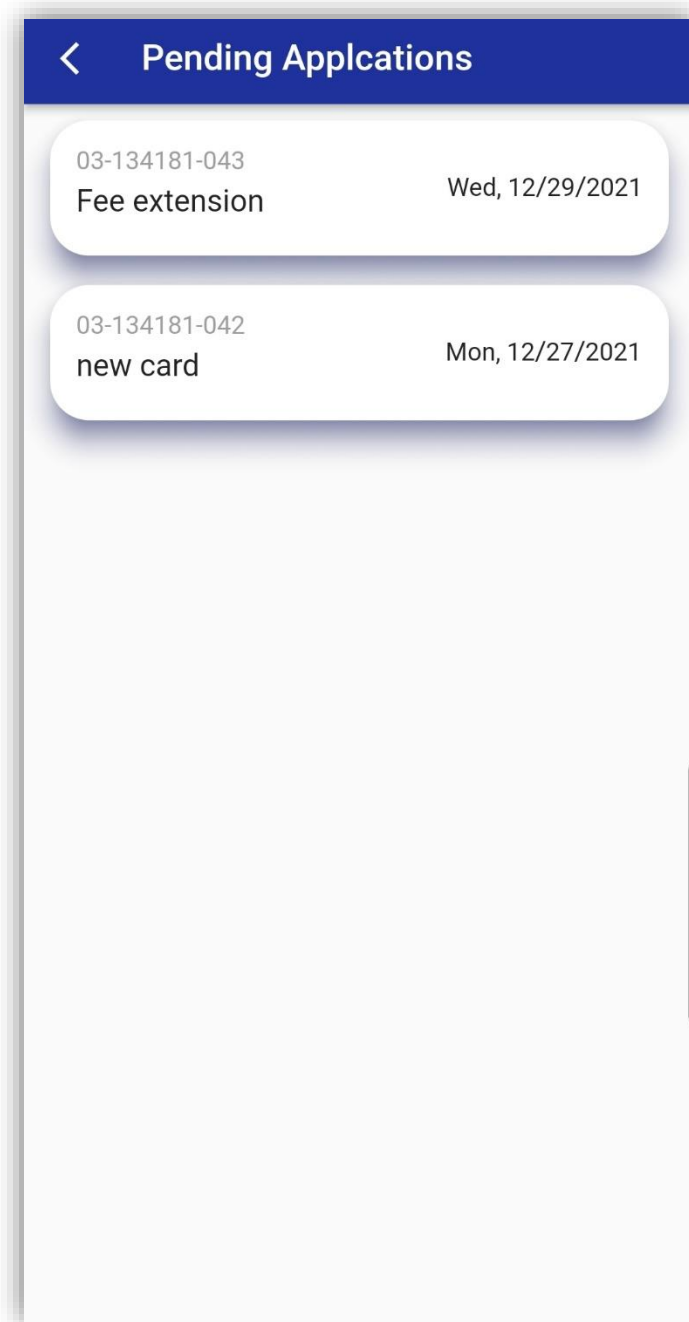


Figure 5.13 Pending applications

5.10 Application details

- Authorized staff can view the application details
- Can add comments
- Can mark the respective department
- Notifications after actions

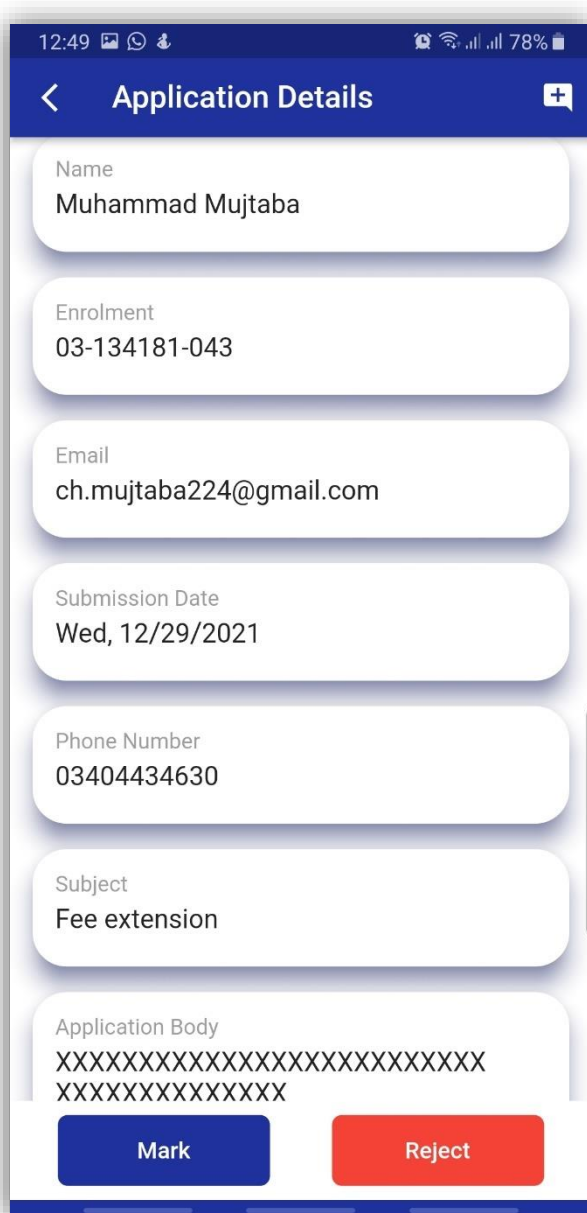


Figure 5.15 Application detail page for faculty

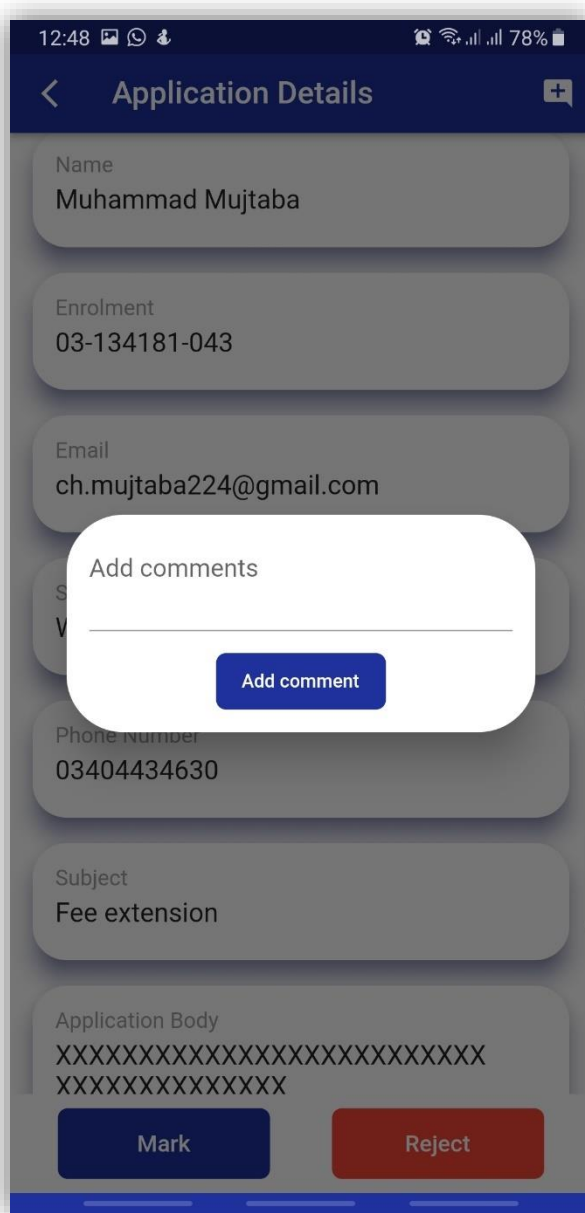


Figure 5.14 Add comment

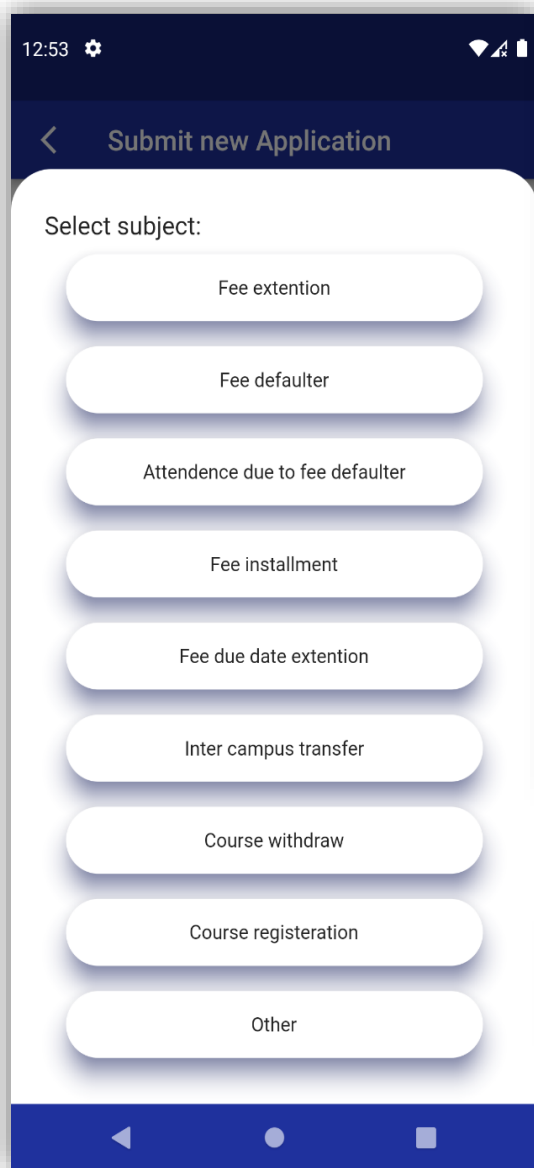


Figure 5.18 Marking application

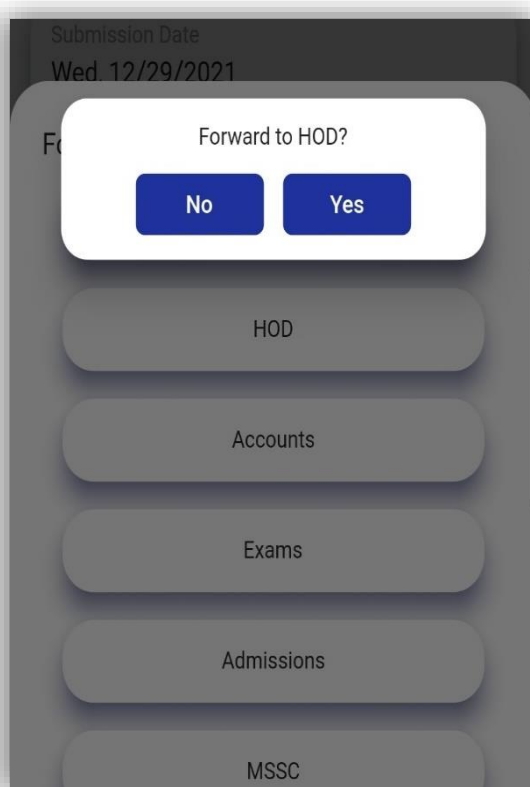


Figure 5.17 Forward confirmation

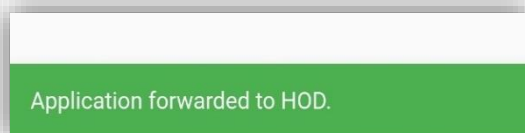


Figure 5.16 Forward successful confirmation

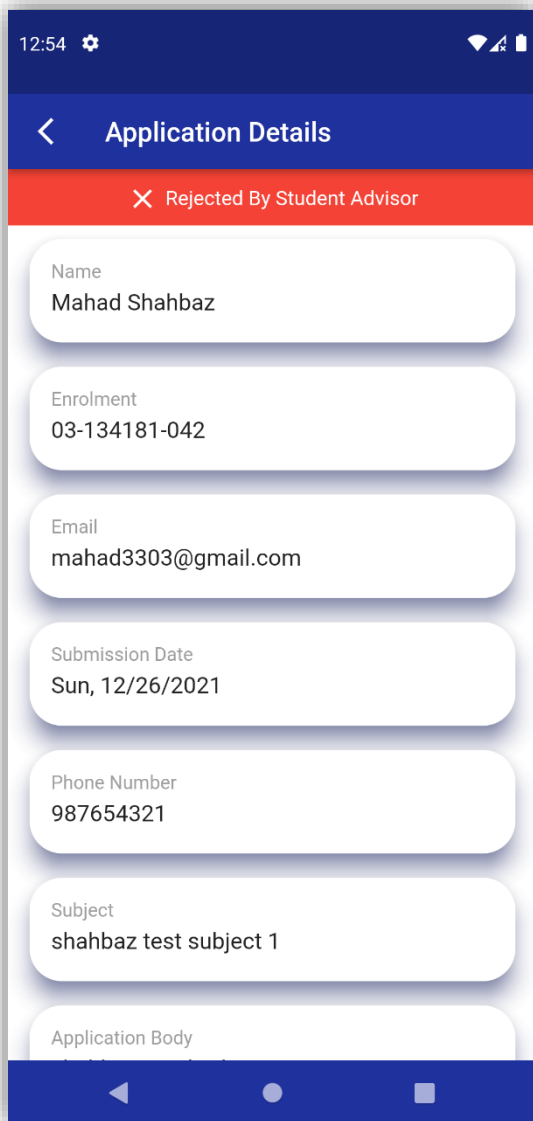


Figure 5.19 Rejected application details

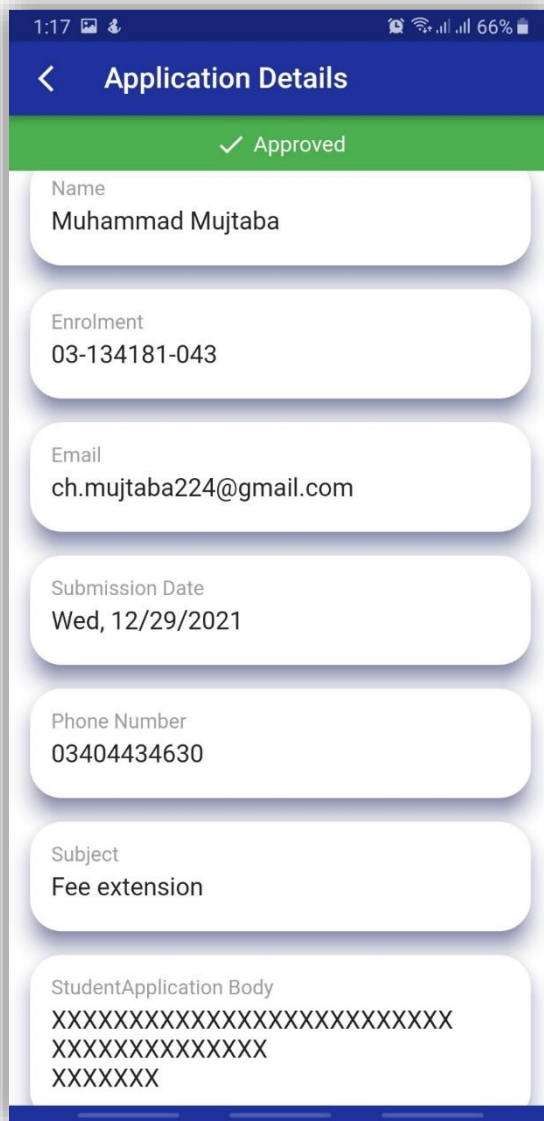


Figure 5.20 Approved application details

5.11 Reports

- Pie chart of according to the status of the applications

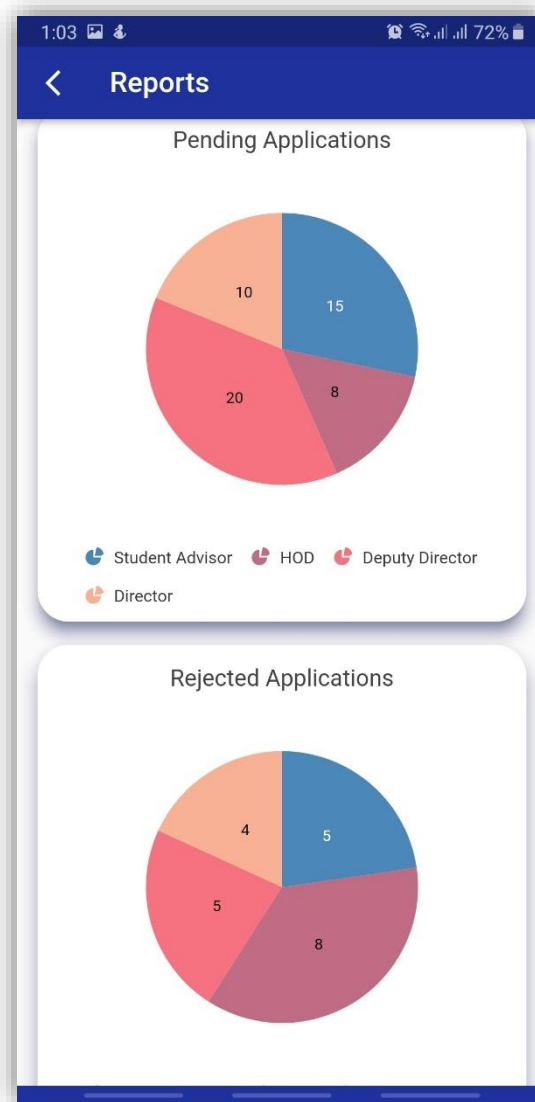


Figure 5.21 Reports page

5.12 History

The history of every application that is submitted is saved for the record-keeping and Authorized staff can search applications of a student.

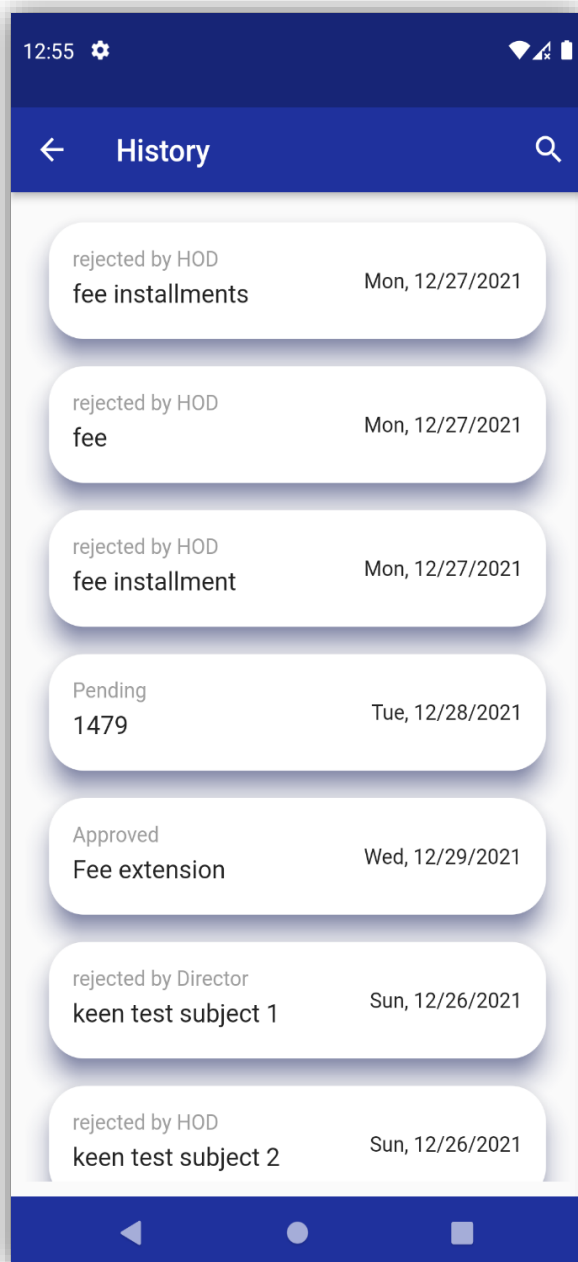


Figure 5.23 History page

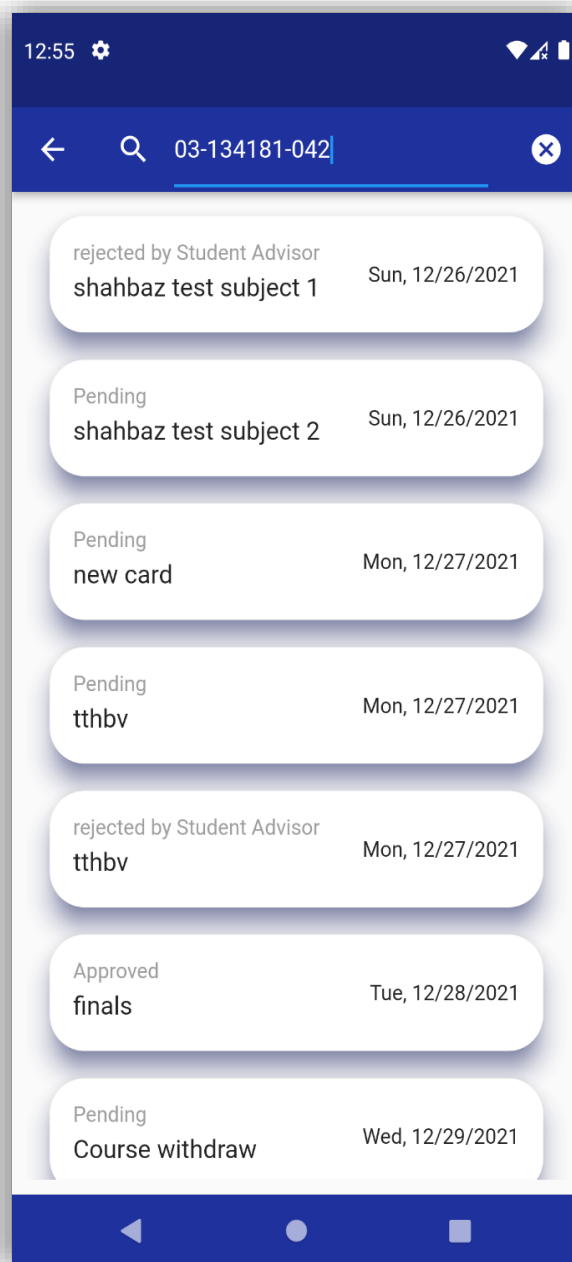


Figure 5.22 History page search bar

5.13 Push notification

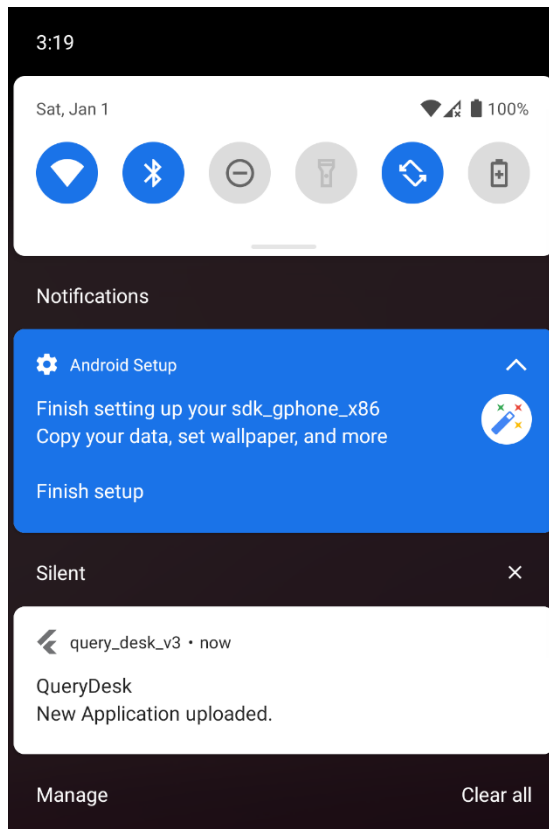


Figure 5.24 New application uploaded notification

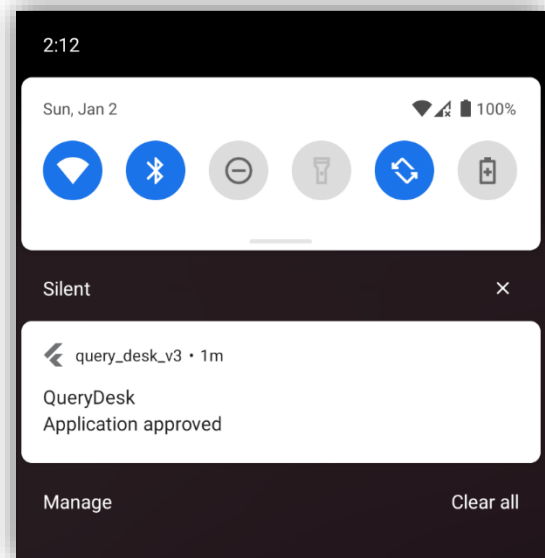


Figure 5.25 Application approved notification

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Project Achievements

Form the verification of plan and after testing the application, the actual output of the QueryDesk is getting good result as manage to reach all the expected output and fulfilled the project objectives. In the end, QueryDesk manages to help the institutions by providing them a platform that will reduce the lengthy and time taking process of application approval. Furthermore, QueryDesk counters the factor which requires timely availability of all authorized personals in the process along with the applicants. By QueryDesk users can perform their intended task remotely.

6.2 Future Work

Some improvements can be made in the future. First and foremost change is according to the business need as every institute has its mechanism of working to give preferences to client-valued functionality we have to change the application UI/UX and mechanism accordingly.

Secondly, we are going to publish a Web App as we had a great advantage of using Flutter for the completion of QueryDesk we can build a Web App with the code based we used. Flutter provides us the opportunity to develop a cross-platform app with a single code base.

6.3 Implementation Issues and Challenges

In this project, we encounter several challenges throughout the project life cycle. First and foremost, development tools and environment need to be determined before the project can proceed. There are many much-integrated development environments such as Android Studio, Visual Studio Code, Netbeans, Eclipse which can be used for the development of QueryDesk. IDEs like Netbeans, Eclipse are outdated and they require extra plugins to integrate with the system while developing a mobile application. After the announcement of Google in 2016 that they are not going to provide the support for these plugins which makes it harder to develop the applications as it is harder for them to integrate with the systems [6]. From Android studio and Visual Studio Code, we use both as both of them are the best tools moreover they have a vast amount of developer community which helps native developers in the hour of need.

On the other hand, as Flutter is new to the market it is still under a lot of development latest version are realizing quickly we had to change our code frequently as the syntax and new features were added to the build

6.4 Conclusion

In this modern era, most educational institutions are using the latest technologies and software but the application submission process is still the same tiring and prolonged as we are still using the same old and outdated process for application management. This is not suitable for this era. QueryDesk has numerous features like status availability, enhanced workflow, etc. so that students get to know at which stage their application is stalled or why their application was rejected. The record of each application submitted by students is kept in cloud-based storage. Hopefully, through this project, we solve this long-awaited upgrade

REFERENCES

Electronic Sources from the Internet:

- [1] “Flutter - Beautiful native apps in record time.” <https://flutter.dev/> (accessed Jun. 18, 2021).
- [2] “Firebase.” https://firebase.google.com/?gclid=CjwKCAjww-CGBhALEiwAQzWxOlbdg3BQdedrIYhzhQuZ7rUpyOharCcPcBzUfGoo6nS7rb1dVbURwxoC7YEQAvD_BwE&gclsrc=aw.ds (accessed Jun. 27, 2021).
- [3] J. Hunt, “Feature-driven development,” *Agil. Softw. Constr.*, pp. 161–182, 2006.
- [4] “android - How to model a Firebase Database in UML diagrams? - Stack Overflow.” <https://stackoverflow.com/questions/54164965/how-to-model-a-firebase-database-in-uml-diagrams> (accessed Sep. 19, 2021).
- [5] “How do I make a relational database on Firebase? - Quora.” <https://www.quora.com/How-do-I-make-a-relational-database-on-Firebase> (accessed Jan. 02, 2022).
- [6] “Google Says Final Goodbye to Eclipse IDE for Android Development -- ADTmag.” <https://adtmag.com/articles/2016/11/07/google-eclipse.aspx> (accessed Oct. 27, 2021).