****

**COMSATS University Islamabad**

**Abbottabad, Pakistan**

**Project Name OPERATING SYSTEM CONCEPS**

**(A Visualizer Tool)**

***By***

**Samra Fida CIIT/FA16-BCS-061/ATD**

**Mariyum Zaib CIIT/FA16-BCS-082/ATD**

**Shafqat Naheed CIIT/FA16-BSE-188/ATD**

***Supervisor*ASHFAQ AHMED**

***Bachelor of Science in Computer Science (2016-2020)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

****

**COMSATS University, Islamabad Pakistan**

**Project Name**

**A project presented to**

**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Computer Science (2016-2020)***

**By**

**Samra Fida CIIT/FA16-BCS-061/ATD**

**Mariyum Zaib CIIT/FA16-BCS-082/ATD**

**Shafqat Naheed CIIT/FA16-BSE-188/ATD**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) was developed by   
**Samra Fida (CIIT/FA16-BCS-061/ATD)** , **Mariyum Zaib (CIIT/FA16-BCS-082/ATD)** and **Shafqat (CIIT/FA16-BSE-188/ATD)** under the supervision of “ASHFAQ AHMED that in (their/his/her) opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

ASHFAQ AHMED

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**EXECUTIVE SUMMARY**

In public places, there is often a need for monitoring people and different activities going on, which can be referred later for many reasons including security. Appointing humans for this task involves many problems such as increased employee hiring, accuracy problem, trust, no proof for later use, and also the fact that a human can remember things till a certain time limit. Talking about the current security system, they use dumb still cameras with a continuous recording facility ir-respective of the fact that any event may happen or not. Moreover they are usually pointing at a specific user defined locations so more than one cameras are required to cover the entire region.

To prevent all these problems from prevailing, the CSCS is developed. It is a surveillance system, which provides solution to many of these problems. It is a stand-alone application which doesn’t require any computer to operate. It monitors different situations using a camera which is able to rotate intelligently based on sensor messages and captures the scene in the form of video or photos later reference as well.

**C**ustomizable **S**urveillance **C**ontrol **S**ystem **(CSCS)** is a surveillance system that can be assigned a sensor type as in our case a heat sensor is used, it works accordingly, rotates the camera upon event detection and perform user defined actions like capturing video and stores them, for the future use.

It is an embedded system consisting of Linux fox kit with embedded a running server application also a camera, USB storage device and a sensor node base station is attached with fox kit. LAN communication is used by user to download the videos and to operate the system manually.

**ACKNOWLEDGEMENT**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Dr. Majid Iqbal Khan” and our Co-Supervisor “Mr. Mukhtar Azeem”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

**ABBREVIATIONS**

|  |  |
| --- | --- |
| **SRS** | Software Require Specification |
| **PC** | Personal Computer |
|  |  |
|  |  |
|  |  |

**TABLE OF CONTENTS**

[**1 Introduction** 11](#_Toc520773734)

[1.1 Brief 11](#_Toc520773735)

[1.2 Relevance to Course Modules 11](#_Toc520773736)

[1.3 Project Background 11](#_Toc520773737)

[1.4 Literature Review 11](#_Toc520773738)

[1.5 Analysis from Literature Review (in the context of your project) 11](#_Toc520773739)

[1.6 Methodology and Software Lifecycle for This Project 11](#_Toc520773740)

[1.6.1 Rationale behind Selected Methodology 11](#_Toc520773741)

[**2** **Problem Definition** 12](#_Toc520773742)

[2.1 Problem Statement 12](#_Toc520773745)

[2.2 Deliverables and Development Requirements 12](#_Toc520773746)

[2.3 Current System (if applicable to your project) 12](#_Toc520773747)

[**3** **Requirement Analysis** 13](#_Toc520773748)

[3.1 Use Cases Diagram(s) 13](#_Toc520773750)

[3.2 Detailed Use Case 13](#_Toc520773751)

[3.3 Functional Requirements 13](#_Toc520773752)

[3.4 Non-Functional Requirements 13](#_Toc520773753)

[**4 Design and Architecture** 14](#_Toc520773754)

[4.1 System Architecture 14](#_Toc520773759)

[4.2 Data Representation [Diagram + Description] 14](#_Toc520773760)

[4.3 Process Flow/Representation 14](#_Toc520773761)

[4.4 Design Models [along with descriptions] 14](#_Toc520773762)

[**5** **Implementation** 15](#_Toc520773763)

[5.1 Algorithm 15](#_Toc520773765)

[5.2 External APIs 15](#_Toc520773766)

[5.3 User Interface 15](#_Toc520773767)

[**6 Testing and Evaluation** 16](#_Toc520773768)

[6.1 Manual Testing 16](#_Toc520773771)

[6.1.1 System testing 16](#_Toc520773772)

[6.1.2 Unit Testing 16](#_Toc520773773)

[6.1.3 Functional Testing 17](#_Toc520773774)

[6.1.4 Integration Testing 17](#_Toc520773775)

[6.2 Automated Testing: 18](#_Toc520773776)

[Tools used: 18](#_Toc520773777)

[**7** **Conclusion and Future Work** 19](#_Toc520773778)

[7.1 Conclusion 19](#_Toc520773780)

[7.2 Future Work 19](#_Toc520773781)

[**8** **References** 20](#_Toc520773782)

**LIST OF FIGURES**

Fig 1.1 Block Diagram 8

Fig 2.1 Use Case Diagram 9

**LIST OF TABLES**

Table 1.1 Table1 8

Table 2.1 Table 2 9

1. **Introduction**

# There are many apps available for learning and the different concepts of OS. As we are making an operating system visualizer, the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals. An operating system (OS) is the program that, after being initially loaded into the computer by a [boot](https://searchwindowsserver.techtarget.com/definition/boot) program, manages all of the other [application](https://searchsoftwarequality.techtarget.com/definition/application) programs in a computer. The application programs make use of the operating system by making requests for services through a defined application program interface ([API](https://searchmicroservices.techtarget.com/definition/application-program-interface-API)). In addition, users can interact directly with the operating system through a user interface such as a command line or a graphical user interface ([GUI](https://searchwindevelopment.techtarget.com/definition/GUI)). An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs.

There are many types of operating systems from which time sharing operating system is a type which uses time slice to share resources between processes.

* 1. **Brief**

This application will allow users to connect each other. First the teacher will make account on this app and study on different topics of operating system and then automatically generated quizzes. The student login their account then start learning of different topic s are mentioned in app. The students or other user s solved the quizzes and the result will be shown on result sheet. It is not necessary to make account only teachers or students. Everyone make account and use this app.

* Android Studio is used for Coding and Designing.
* Sequential methodology is being used.

Following table shows the tools which we used in this app and also these tool and technologies helped us to clear the presentations and work required for the project completion.

Table 1.1 shows details of tools and technologies used.

|  |  |  |
| --- | --- | --- |
| **Tools** | **Version** | **Rationale** |
| Android Studio | 2019 | Coding |
| MS Power Point | 2016 | Presentation |
| MS Word | 2016 | Documentation |
| Star UML | 2016 | Diagrams |

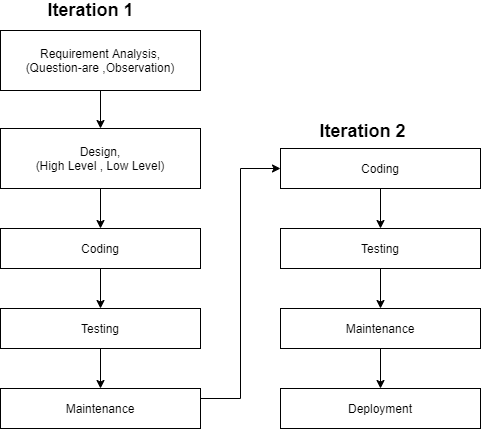
* 1. **Relevance to Course Modules**

Our final year project is an application which runs on android device. Our project is related to Operating system.

* 1. **Analysis from Literature Review (in the context of your project)**
  2. **Methodology and Software Lifecycle for This Project**

For this app we will use the process model called “**Iterative model**”.

It can be started with some of the software specification and development of the first version of application. After the first version, the new version of application can be created with new iteration; if required.



2. **Problem Definition**

This chapter discusses the precise problem to be solved. It should extend to include the outcome.

3. 1. **Problem Statement**

# As we are making an operating system visualizer, the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals.

The purpose of this application is to help the students understanding the concepts of Operating system. This application will be equally useful for the students of masters program. This application will help student to understand the functionality of operating system.  As operating system is a system level software in which many modules are working simultaneously. An operating system is managing your current running foreground and background processes. It is managing your files and storage devices. The purpose of this project is to help student visualize the concepts of operating systems.

As in previous applications ,they don’t have visualization concepts and quiz generation module together, so for the better understanding and clarification of the concepts of operating system such as different scheduling algorithms ,deadlocks etc this applications is helpful for students as well as for teachers. The topics that are explained will be further explain through visualization ,so that students can learn these difficult topics easily . Students will also be able to visualize different Process scheduling, memory management techniques. It is an interesting module in which some students can understand different processes such as FCFS,SJF etc and also can practice by itself by giving different random inputs. Another important thing is that it has the record of user login/history, so if they want to use this application they don’t need to login again. The quiz which user attempt will also be saved in database. Students can learn detailed notes that are present in learning section , then after they can take self-test by attempting quiz present after every topic. Quiz will generate and at the end it will display marks. It is useful application as students can test their knowledge and make their concepts stronger. This module students will be able to assess themselves. User can give quiz of different topics in order to monitor their progress. The topics that are discussed contains mcq’s type questions so that user can judge by itself. We will make these mcq’s from the topics that are discuss in learning module. For every topics, there consist of having some easy, medium and hard multiple choice.

* 1. **Deliverables and Development Requirements**

The users cannot use this app without internet. Internet will be compulsory to connect to each other without internet the user cannot signup or login their accounts. Updated version also compulsory.

* Android phones are mandatory. As this application is built for most operating system used in this world so to make it more prosper Android application was good option.
* A PC with Android Studio to develop an application. Android studio is the tool in which all the coding will take place to develop the application. Many external APIs are supported by Android studio.
* Simple database along with SQL is used to store data.

**Current System (if applicable to your project)**

Following are some systems which are near similar to our project.

**Operating System – OS**

The app is a complete free handbook of Operating System which covers important topics, notes, materials, news & blogs on the course. Download the App as a reference material & digital book for computer science engineering & software engineering programs & tech degree courses.

This useful mobile App lists 125 topics with detailed notes, diagrams, equations, formulas & course material, the topics are listed in 5 chapters. The app is must have for all the engineering science students & professionals.

The app provides quick revision and reference to the important topics like a detailed flash card notes, it makes it easy & useful for the student or a professional to cover the course syllabus quickly before an exams or interview for jobs.

Track your learning, set reminders, edit the study material, add favourite topics, share the topics on social media. You can also blog about engineering technology, innovation, engineering startups, college research work, institute updates, Informative links on course materials & education programs from your smartphone or tablet or at http://www.engineeringapps.net/.

Use this useful engineering application as your tutorial, digital book, a reference guide for syllabus, course material, project work, sharing your views on the blog.

Some of the important topics Covered in this app.

**Operating System - All In On**

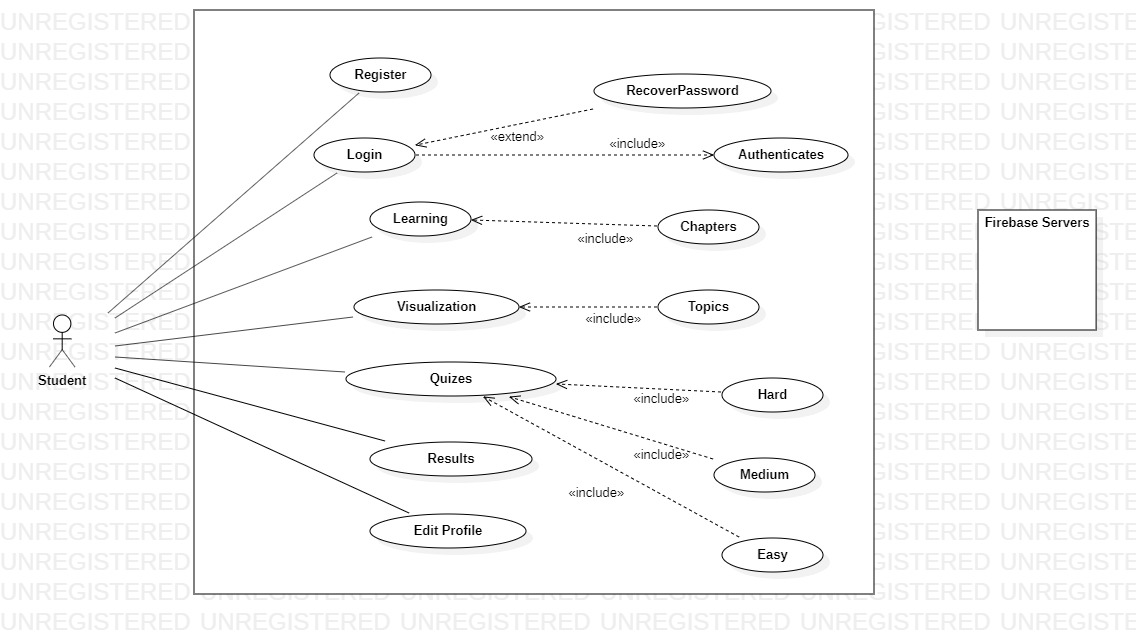
“OPERATING SYSTEM - ALL IN ONE” app provides an environment to learn and prepare in CONCEPT OF OPERATING SYSTEM from anywhere, at any time and beyond the limits. This "OPERATING SYSTEM - ALL IN ONE" is for all kinds of preparation like GATE, UNIVERSITY EXAM, COMPETITIVE EXAM. And especially for students of BE, Diploma, MCA, BCA students. This app is aimed to grow your knowledge and quick reference. An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. All computer programs, excluding firmware, require an operating system to function.

1. **Requirement Analysis**

The following parts of Software Requirements Specification (SRS) report should be included in this chapter.

2. 1. **Use Cases Diagram(s)**

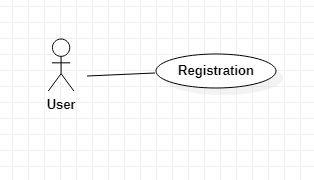
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



* 1. **Detailed Use Case**

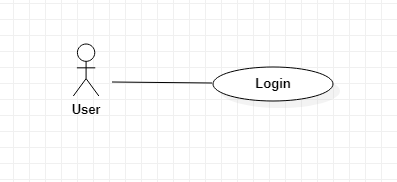
**User Module:**

|  |  |
| --- | --- |
| Use Case ID | UC-1 |
| Use-Case Name | Registered user |
| Actors | Primary actor(user) |
| Description | User can request to registered him and the server registered the user.  User create account. |
| Trigger | User want to be register in application. |
| Dependency | Include Check user name or password. |
| Preconditions | To installed the visualizer tool in your system. |
| Post conditions | When registered the user then he can login in the app and used it. |
| Normal Flow  Or  Main Sequence | Enter the required data into input fields then press sign up button.  Data added to the database successfully. |
| Alternative Flows | 1a. User enter the required data.  1. If any mandatory data is not entered.  1. The error message will be displayed. |
| Special Requirements | User must have android mobile phone. |



**Login Module:**

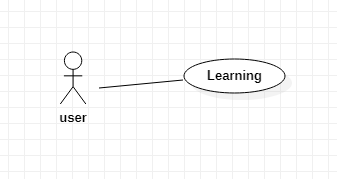
|  |  |
| --- | --- |
| Use Case ID: | UC-2 |
| Use Case Name: | Login |
| Actors: | Primary(user) |
| Description: | User can request to login. |
| Trigger: | User wants to be login in application. |
| Dependency: | Include Check user name or password. |
| Preconditions: | Open the application and click on login button. |
| Post conditions: | Data is saved into database, database updated. |
| Normal Flow:  Or  Main Sequence | Enter the required data into input fields then press sign up button.  Data added to the database successfully. |
| Alternative Flows: | 1a. User enter the required data  1. If any mandatory data is not entered  1. The error message displayed. |
| Special Requirements: | User must have android mobile phone for this app. |



**Learning Module:**

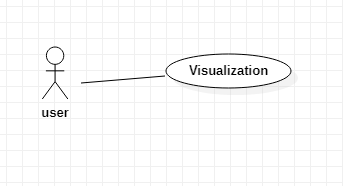
*Table 3.3 Seller Sign in*

|  |  |
| --- | --- |
| Use Case ID: | UC-3 |
| Use Case Name: | Learning |
| Actors: | Primary(user) |
| Description: | User learn different topics into the app. |
| Trigger: | User wants to be learn data from app. |
| Dependency: | Include Check user name or password. |
| Preconditions: | Signed up or logged in. |
| Post conditions: | Go to home page.  Take quizzes of relates chapters. |
|  |  |
| Alternative Flows: | Username or password is incorrect.  The error message displayed. |
| Special Requirements: | Seller must have android mobile phone |



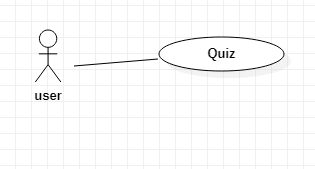
**Visualization Module:**

|  |  |
| --- | --- |
| Use Case ID: | UC-5 |
| Use Case Name: | visualization |
| Actors: | Primary(user) |
| Description: | animation |
| Trigger: | The user wants to visualization. |
| Dependency | Include Check user name or password. |
| Preconditions | User must learn the concepts of OS. |
| Post-conditions | User takes quizzes. |
| Normal Flow  Or  Main Sequence | If user not know the concepts of OS. They cannot take quizzes. |
| Alternative Flows | If any problem comes the user go to home page. |
| Special Requirements | User must have android mobile phone |



**Quiz Module:**

|  |  |
| --- | --- |
| Use Case ID: | UC-6 |
| Use Case Name: | Quiz |
| Actors: | Primary(user) |
| Description: | User takes quiz. |
| Trigger: | Seller wants to be different types of quiz e.g. hard type, medium or normal. |
| Dependency: | Include Check user name or password. |
| Preconditions: | User must be login into the system.  Know the concepts of OS.  Then take quiz. |
| Post-conditions: | Start the quiz.  Check scores. |
| Normal Flow:  Or  Main Sequence | Take the quiz and their marks. |
| Alternative Flows: | User name or password not correct.  Error message show. |
| Special Requirements: | User must have android mobile phone |



**Record Module:**

|  |  |
| --- | --- |
| Use Case ID: | UC-7 |
| Use Case Name: | Record |
| Actors: | Primary(user) |
| Description: | All results or records show. |
| Dependency: | Include Check user name or password. |
| Preconditions: | User must be login into the system and learn and take quizzes. |
| Post conditions: | Show Records. |
| Special Requirements: | Seller must have android mobile phone. |

* 1. **Functional Requirements**

The functionalities of the application will be:

**Signing up:**

This app will provide the opportunity to sign up. Signing up will save the details of user.

**Editing profile:**

Once a person will sign-up and login, they cannot change their information that is there email, there user name, password.

**Learning (Detailed Notes):**

The User will once login in successfully and have mobile phone then the user learn the different topic s or chapters of OS.

**Quiz:**

The user will after login and study the OS concepts then solved the quiz. Quizzes are three types’ hard medium or normal.

**Visualization:**

When the user will take quiz then visualization module start. After visualization all results are shown into the table.

* 1. **Non-Functional Requirements**

**Startup Time (Performance)**

Application will start in few (1 to 2) seconds but it depends upon the internet connection of person who uses it.

**Response Time**

Application response time should be fast so that the home page will be quickly seen within a limited time after login or signup.

**Safety and Security Requirements**

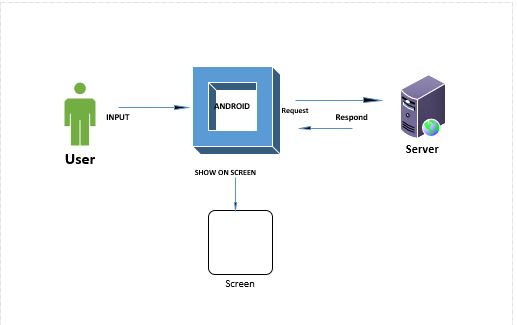
* Security requirements are important factors in this system as classified data will be stored in the database.
* User validation will be done during login to ensure that the user is valid and that the user only has access to his or her permission data.
* It checks whether a given user name or password is valid or not and then it gives access the user to use the app. False record input by the user is checked by the system and system does not give access to the user to use the app.

1. **Design and Architecture**

The following parts of Software Design Description (SDD) report should be included in this chapter.

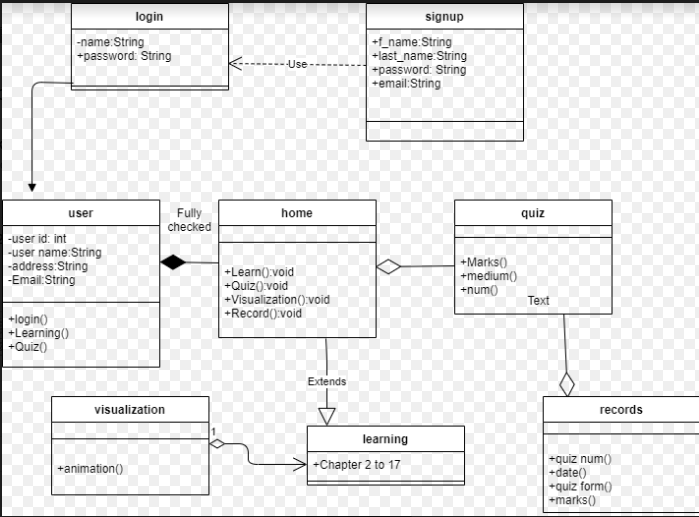
**System Architecture**

Figure will show the Design Architecture of the system.

****

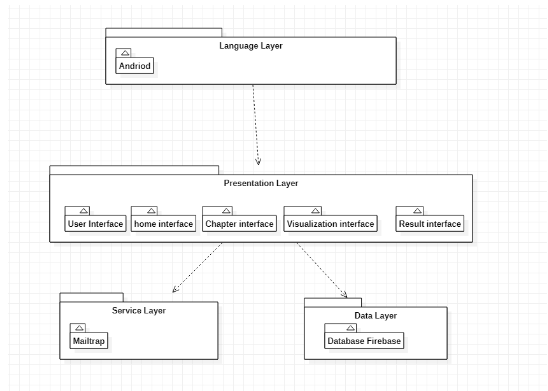
* 1. **Data Representation(Diagrams)**

**Class Diagram**

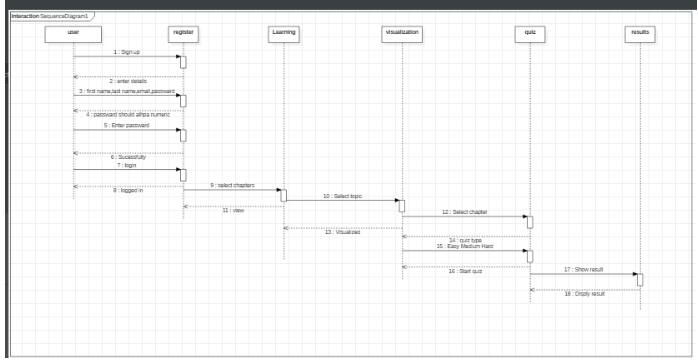




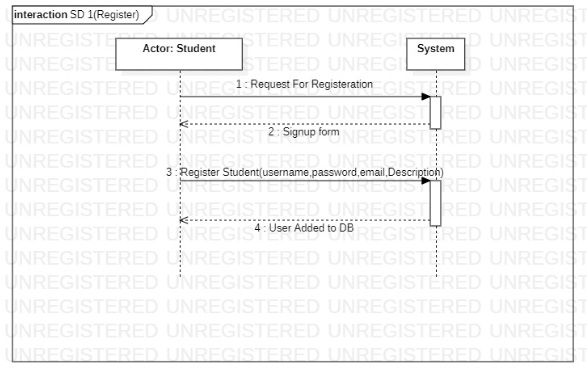
**Package diagram**



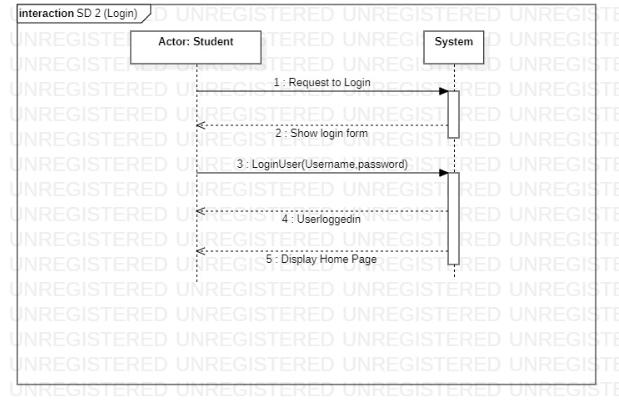
* + 1. **Sequence Diagrams**

****

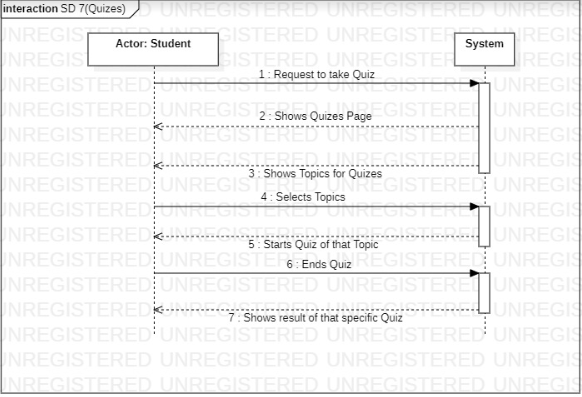
**Sequence Diagram registration**

****

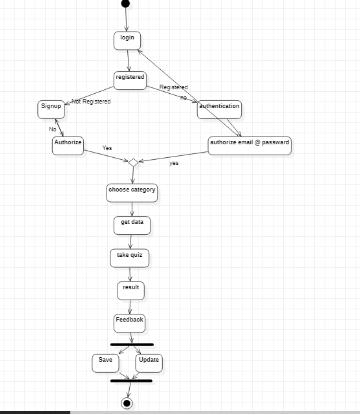
**Sequence Diagram login**

****

**Sequence Diagram Student**

******Process Flow/Representation:**A process flow diagram is a diagram commonly used in process to indicate the general flow of processes

**Activity Diagram**



* 1. **Design Models [along with descriptions]**

In this application, object-oriented approach is used as a design method. Hence, it is easier to implement the project and add possible future features. Furthermore, multi-layered system architecture is used. Layers will help modularity, security and adaptability of the software. With object-oriented design and multi-layered architecture, portability and integrity between components will be improved.

* To store User information we have attached database with the system.

**Data Dictionary**

Below table will show the Data Dictionary used into the system.

|  |  |
| --- | --- |
| **Iterative Model** | Iterative process models with focus on process in different iteration with adaptability and user will be satisfied. |
| **Object-Oriented** | Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields, and code, in the form of procedures. |
| **Database** | Simple data base is used to store the user information. |

1. **Implementation**

This chapter will discuss implementation Operating system concepts: A visualization tool Application into the system.

1. 1. **Algorithm**

Algorithms are most important steps in any system you are going to design implement. For most activities’ algorithms are used are discussed in this chapters.

**Pseudo code for Login Activity:**

* Declare user name as a String.
* Declare password as a numeric or string.

**Pseudo code for quiz class:**

* Declare selected topic.

**Pseudo code for visualization class:**

* Declare visualization of the topic.

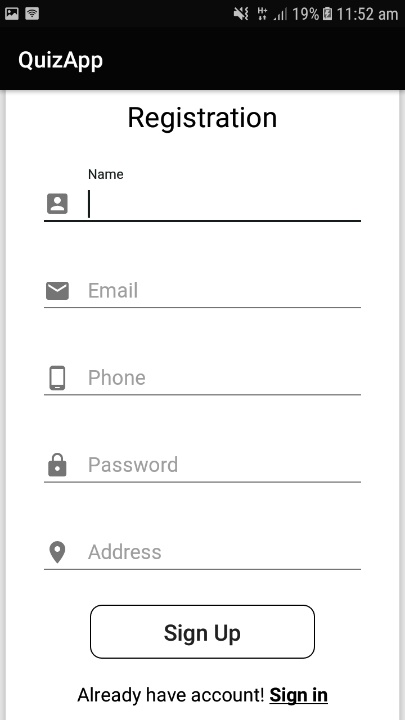
**Pseudo code for records class:**

* Declare records as a numeric.
* Declare show records() function.

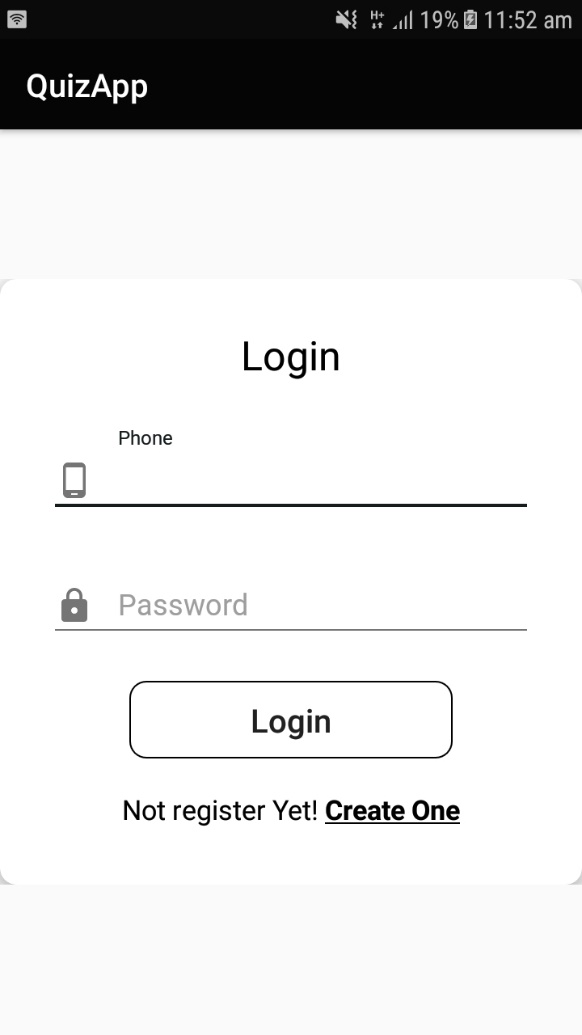
**User Interface**

User interface will show the design and look of the application made in the end of the project.

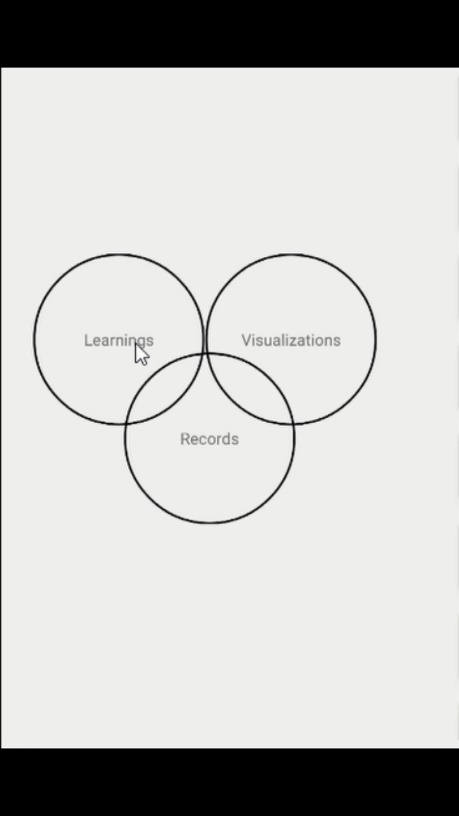
**Registration page:**



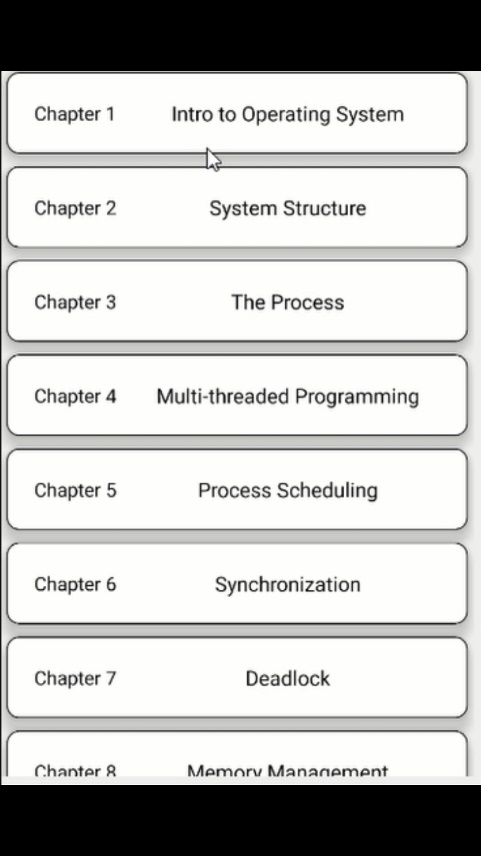
**Login page**

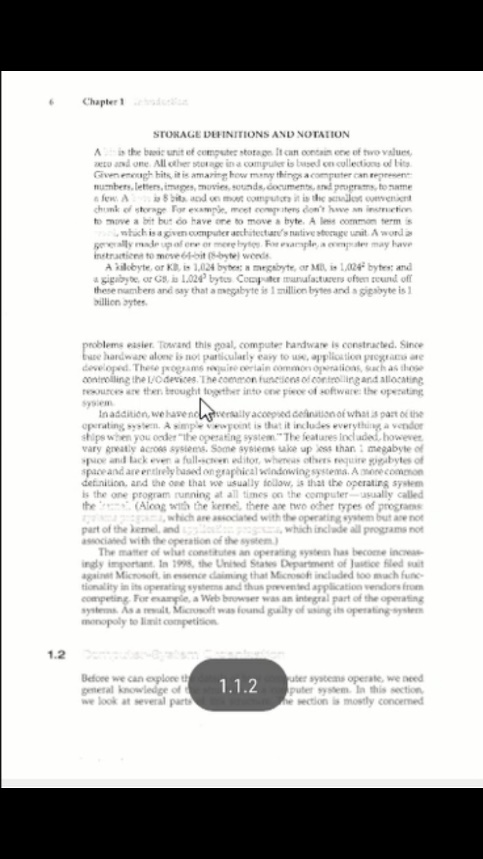


**Home page**

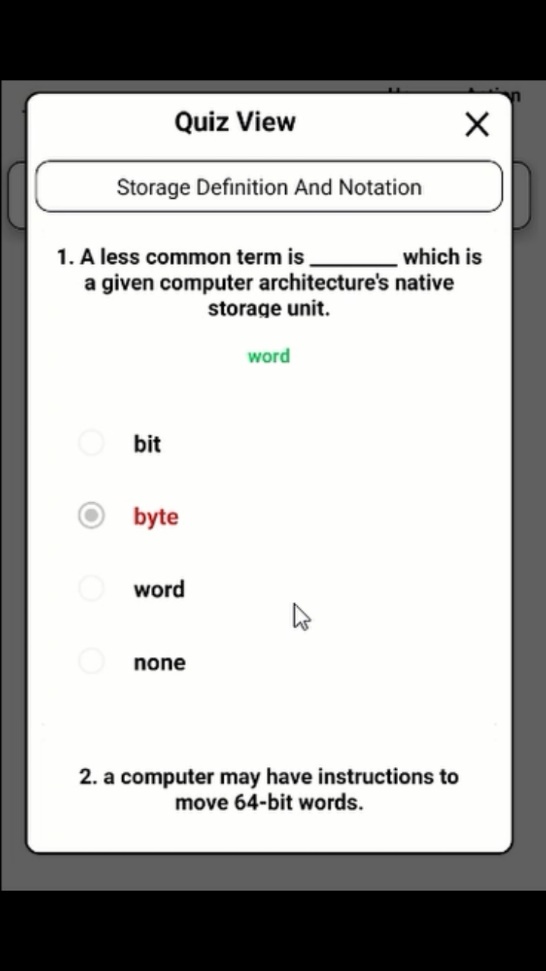
****

**Learning**

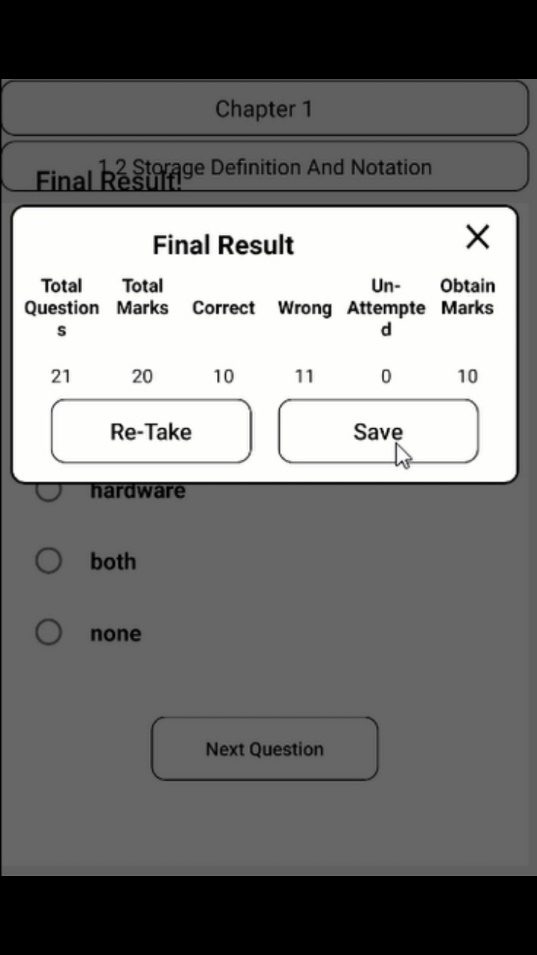




**Quiz**



**Records**



1. **Testing and Evaluation**

This chapter may include different type of testing that we perform on our application

3. 1. **Manual Testing**

In manual testing we will test this application to ensure that it performs work correctly or not by perform unit testing, system testing, functional testing and integration testing.

1. * 1. **System testing**

A visualizer tool: This is successfully developed testing has been performed so we ensure that the system will working as intended. This app will meet the requirements as stated in requirement analysis section.

* + 1. **Unit Testing**

Each unit of application is tested under different test cases.

**Unit testing 1:** login:

**Testing Objective:** To ensure that login page work successfully.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Verify user login after click on the ‘log in’ button on log in page form with correct input data | User name or password: Valid | Successfully log into Home page of application. | Pass |
| 2. | Verify user login after click on the ‘login’ button on login in page form with wrong input data | User name or password:  In-Valid | User will not be logged in to Home page of application | Pass |

**Unit Testing 2:** Edit user name, password

**Testing Objective:** To ensure the edit profile form is working properly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Click on setting and edit user name, password | Enter new name , password | Username, password successfully update. | Pass. |
| 2. | Click on setting and edit user name and Email | Enter new name | Username, successfully | Pass |
| 3. | Click on setting and edit user name and Password | Enter new name and Password | Username and password successfully updated | Pass |

* + 1. **Functional Testing**

The functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

* **Functional Testing 1:** Sign-up with Username and Valid email

**Objective**: User Signed up Successfully

*Table 5.3: Login Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Verify user sign after click on the ‘Sign up’ button on Sign up page form with correct input data | User name or Email: Valid | Successfully log into Home page of application. | Pass. |
| 2. | Verify user sign after click on the ‘Sign up’ button on Sign up page form with wrong input data | User name or email: In-Valid | User will not be logged in to Home page of application | Pass |

**Functional Testing 2:** Login with Username and Password

**Objective**: User Login Successfully.

Below table will show the testing of login of user

*Table 6.4. Login of user*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Verify user sign after click on the ‘Log in’ button on Log in page form with correct input data | User name and Password | Successfully log into Home page of application. | Pass. |
| 2. | Verify user Login after click on the ‘Log in’ button on Login page form with wrong input data | Phone  Number/Username/Password: In-Valid | User will not logged in to Home page of application | Pass |

**Functional Testing 3:** Quiz

**Objective**: User takes quiz successfully.

Below table will show the testing of take quiz.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Verify user click on button start quiz from quiz section. | Quiz will be shown. | Successfully Started. | Pass. |
| 2. | Verify user click on button start quiz from quiz section. | Quiz will not be shown. | User will not started the quiz. | Pass |

* + 1. **Integration Testing**

Below table will show the integration testing of sign in

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | login in as user | Enter Valid. username or password | User successfully logged in to home page of the application. | Pass |

* 1. **Automated Testing:**

Different test cases made and against each test case application is checked.

* + 1. **Tools used:**

**Appium** tool for automated testing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool Name** | **Tool Description** | **Applied on** | **Results** |
| **Appium** | **Appium**. An open-source mobile test automation tool to test Android and iOS applications. | 1. Signup 2. Login 3. Quiz 4. Upload Records | Pass |

**Conclusion and Future Work**

This chapter concludes the project and highlights future work.

* 1. **Conclusion**
* This application is for students and also for teachers to understand the concepts of operating system.
* Our target is make easy and helpful for students to learn ,visualize and test their knowledge through quizzes.
* The purpose of this application is to help the students understanding the concepts of Operating system. This application will be equally useful for the students of masters program
* This application will help student to understand the functionality of operating system.  As operating system is a system level software in which many modules are working simultaneously. An operating system is managing your current running foreground and background processes. It is managing your files and storage devices.
* The purpose of this project is to help student visualize the concepts of operating systems.
* In learning module students will be able to study different Operating systems topics. Different concepts of operating system will discuss with detail information. Students will learn with ease. Detail notes will be provided to understand better.
* The topics that are explained will be further explain through animations ,so that students can learn these difficult topics easily .
* students will also be able to visualize different Process scheduling, memory management techniques.
* It is an interesting module in which some students can understand different processes such as FCFS,SJF etc and also can practice by itself by giving different random inputs.
* At the end of each topic, students will be able to assess themselves. User can give quiz of different topics in order to monitor their progress. The topics that are discussed contains mcq’s type questions so that user can judge by itself.
* We make these mcq’s from the topics that are discuss in learning module. For every topics, there consist of having some easy, medium and hard multiple choice.
  1. **Future Work**
* This application will users to make their concepts stronger, to judge themselves by attempting quiz.
* Detailed notes are present topic wise , so after detail learning, user can attempt quiz to check that how much they understand.
* As there are some scheduling algorithms, for better understanding, visualization is performed.
* User can learn by inputs process number as they want, so process will run according to the input.
* User profiling is one of the best approach , in which user only login once, and use this application.
* Records are also saved in database, marks of attempted quizzes.
* Requiz option will also be available.

**References**

References to any book, journal paper or website should properly be acknowledged. Please consistently follow the style. The following are few examples of different resources i.e. journal article, book, and website.

* 1. Lyda M.S. Lau, Jayne Curson, Richard Drew, Peter Dew and Christine Leigh, (1999), Use Of VSP Resource Rooms to Support Group Work in a Learning Environment, ACM 99, pp-2. **(Journal paper example)**
  2. Hideyuki Nakanishi, Chikara Yoshida, Toshikazu Nishmora and TuruIshada, (1996), FreeWalk: Supporting Casual Meetings in a Network, pp 308-314 **(paper on web)** http://www.acm.org/pubs/articles/proceedings/cscw/240080/p308-nakanishi.pdf
  3. Ali Behforooz& Frederick J.Hudson, (1996), Software Engineering Fundamentals, Oxford University Press. Chapter 8, pp255-235. **(book reference example)**
  4. Page Author, Page Title, http://www.bt.com/bttj/archive.htm, Last date accessed**. (web site)**
  5. <https://online.visual-paradigm.com/diagrams/features/uml-tool/(Web> Site)
  6. <https://codex.cs.yale.edu/avi/os-book/(Book)>
  7. <https://play.google.com/store/apps/details?id=com.bitsofcomputer.pradip.operatingsystem&hl=en(Web> Site)
  8. <https://play.google.com/store/apps/details?id=com.bitsofcomputer.pradip.operatingsystem&hl=en>
  9. <https://play.google.com/store/apps/details?id=com.faadooengineers.free_operatingsystemcompscengg&hl=en>