Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Purpose

This document is the Software Requirements Specification (SRS) for the “School of Programming”. It contains detailed functional, non-functional, and supports requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as the official means of communicating about the user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents. Software architectural design has been followed in “School of Programming” At the beginning of the architectural design the context diagram of the “School of Programming” was defined. Then the archetype of the project is described. And after the finding the archetypes the components and the corresponding classes are defined.

1.2 Intended audience

This SRS is intended for several audiences, including the customer as well as the project managers, designers, developers, and testers.

- The customer will use this SRS to verify that the developer team has created a product that is acceptable for the customer.
- The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the development team is on track during development of the system.
- The designers will use this SRS as a guideline for creating the system’s design.
- The designers will continuously refer back to their SRS to ensure that the system they are designing will fulfill the customer’s needs.
- The developers will use this SRS as a basis for developing the system’s functionality.
- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are being completed, the testers will run their tests on the software to ensure that the software fulfills the requirements documented on the SRS. The testers will again run their tests on the entire system when it is completed and more use of that all requirements documented in the SRS have been fulfilled.

Chapter 2

Inception

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and how is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. To establish the groundwork, I have worked with the Factors related to the inception phases:

- Identifying Stakeholders
- Asking the First Questions
- Recognizing Multiple Viewpoints
- Working Towards Collaboration

2.1.1 Identifying stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in organization that may be affected by its installation. To identify the stakeholders, we consulted with Admin and asked him following questions:

- Who is paying for the project?
- Who will be using the project outcomes?
- Who gets to make the decisions about the project (if this is different from the money source)?
- Who has resources I need to get the project done?
- Whose work will my project affect? (During the project and also once the project is completed)

Concluding thoughts on Stakeholders, we identified following stakeholders for our automated School of Programming project:

1. **Admin:** The admin has the administrative power to manage the overall system.

2. **User:** The user can view the tutorial, participate in quiz. There is an opportunity for guest users to go through the content only, but they can’t participate in the quiz.

3. **Developers:** We selected developers as stakeholder because they develop this system and work for further development. If any system interruption occurs, they will find the problem and try to solve it.

2.1.2 Asking the First Questions

We set our first set of context-free questions focusing on the users and other stakeholders, overall project goals and benefits. The questions are mentioned above. This questions helped us to identify all stakeholders measurable benefit of the successful implementation and possible alternatives to custom software development. Next set of question helped us to gain a better understanding of problem and allows the customer to voice his or her perception about the solution. The final set of question focused on the effectiveness of the communication activity itself.

2.1.3 Recognizing multiple viewpoints

1. Admin:

- Web-Based Interfaces
- Restrict access to functionality of the system based upon user roles.
- The application can be accessed from any computer that has Internet access.

2. User:

- Allow the system to be accessed via the Internet.
- Easy Access
- Allows valid users to view contents online by logging into the system.

2.1.4 Working towards collaboration:

Every stakeholder has their own requirements. We followed following steps to merge these requirements:

- Identify the common and conflicting requirements
- Categorize the requirements
- Take priority points for each requirements from stakeholders and on the basis of this voting prioritize the requirements
- Make final decision about the requirements.

Common requirements:

- Web-Based Interfaces
- The application can be accessed from any computer that has internet access

Conflicting Requirements:

We found some requirements conflicting each other. We had to trade-off between the requirements.

- Easy access and Strong Authentication.
- Allow any user to enter the system and allow valid user to use the system.

Final Requirements:
We finalized following requirements for the system by categorizing and prioritizing the requirements:

- Web-based interface.
- Allow valid users to sign up, sign in and sign out.
- Restrict access to functionality of the system based upon user roles.
- Only admin can add, edit or delete any content.
- General users can learn the content and participate in the exam.
- Guest users can view the contents only.

2.1.5 Conclusion:
Inception phase helps us to establish basic understanding about School of Programming; identifies the people who will be benefited if School of Programming becomes automated, defines the nature of the School of Programming software and establishes a preliminary communication with our stakeholders.

Chapter 3

Elicitation

3.1 Introduction

Elicitation is a task that helps the customer to define what is required. To complete the elicitation step I face many problems like problems of scope, problems of volatility and problems of understanding. However, this is not an easy task. To help overcome these problems, I have worked with the eliciting requirements activity in an organized and systematic manner.

3.2 Eliciting Requirements:

Unlike inception where Q&A (Question and Answer) approach is used, elicitation makes use of a requirements elicitation format that combines the elements of problem solving, elaboration, negotiation, and specification. It requires the cooperation of a group of end-users and developers to elicit requirements. To elicit requirements, I completed following four works.

1. Collaborative Requirements Gathering.
2. Quality Function Deployment.
3. Usage Scenarios.
4. Elicitation work products.

3.3 Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. The following steps have been completed to do it.

- The meetings were conducted with the project supervisor.
- The problems of current tutorials based web application have been discussed.
- At last the final requirement list from the meetings have been selected.

3.4 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to this project the following requirements are identified by a QFD.

3.4.1 Normal Requirements

Normal requirements consist of objectives and goals that are stated during the meeting with the customers. Normal requirements are:

1. Accessible via the Internet.

2. Allow users to sign up, sign in and sign out.
3. Restrict access to functionality of the system based upon user roles.
4. Allow admin to add, edit, search and delete any content.
5. General users can view the contents of tutorials and participate in the exam.
6. Guest user can only view the contents.

3.4.2 Expected Requirements

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause for dissatisfaction.

1. Admin will set date for schedule quiz and add questions for both schedule and practice quizzes.
2. Admin will send the result of scheduled quiz to the general users via mail.
3. The progress of any valid user must be stored in database.
4. The general users can see the result of any time quiz instantly.
5. The system will enable the admin to change or update any information.

3.4.3 Exciting requirements

These requirements go beyond the customer's expectations and prove to be very satisfying when present.

- The user interface should provide appropriate error messages for invalid input.
- The user interface should follow standard web practices such that the web interface is consistent with typical internet applications.
- The system's configuration shall be documented and updated as changes to the system are made due to patches, new releases, new Changes etc.

3.5 Usage Scenario

School of Programming is a web application which will provide tutorials of different programming and web development languages. This application will be used mainly by the admin and general user.

Admin can add, search, edit or delete the content of this application through content management system. General users will go through the content to learn them & participate in quiz. There is opportunity for guest users to go through the content only, but they can't participate in the quiz.

Admin will add question for quiz exam. A general user needs to register by providing necessary information such as name, username, email, password etc. After registration process, a user can sign in, view the tutorials and participate in the quiz. A general user can participate in schedule quiz and practice quiz. The schedule quiz will be taken after a certain time from the registration. The admin will let the general users know about the schedule quiz date and topic via email. For more practice, a user can participate in the practice quiz at any time. There will be multiple choice and true/false type questions. After giving the quizzes, the users will get their marks.

3.6 Elicitation work product

The output of the elicitation task can vary depending on size of the system or product to be built. Our elicitation work product includes:

- A statement of our requirements for automated “School of programming”.
- A bounded statement of scope for the proposed system.
- A list of customers, users and other stakeholders who participated in requirement specification.
- Set of usage scenarios.
- Description of the system’s technical environment.

Chapter 4

Scenario based Modelling

This chapter describes scenario based modeling of School of Programming.

4.1 Definition of use case

A use case is a software and system engineering term that describes how a user uses a system to accomplish a particular goal. A use case acts as a software modeling technique that defines the features to be implemented and the resolution of any errors that may be encountered.

Use cases define interactions between external actors and the system to attain to particular goals. There are three basic elements that make up a use case:

1. Actors: Actors are the type of users interact with the system.
2. System: Use cases capture functional requirements that specify the intended behavior of the system.
3. Goals: Use cases are typically initiated by a user to fulfill goals describing the activities and variants involved in attending the goal.

Use cases are modeled using unified modeling language and are represented by ovals containing the names of the use case. Actors are represented using lines with the name of the actor written below the line. But here, we use the combination of lines and a circle to represent each actor. To represent an actor's participation in a system, a line is drawn between the actor and the use case. Boxes around the use case represent the system boundary.

There are two types of actor:

1. Primary actor.
2. Secondary actor.

Primary actor: Primary actor refers who is directly involved with the system in order to achieve required function and benefit from the system. They interact directly and frequently with the software.

In our proposed system, admin and general user are primary actor.

Secondary actor: Secondary actor refers who is indirectly involved with the system but necessary to support the system so that system can perform its functionality without any hinder. Secondary actor either produces or consumes information.

4.2 Use case diagram

The use case scenario to use case diagram, description, activity diagram and swim lane diagram is elaborated. Here is the use case diagram of level-1 for School of Programming:

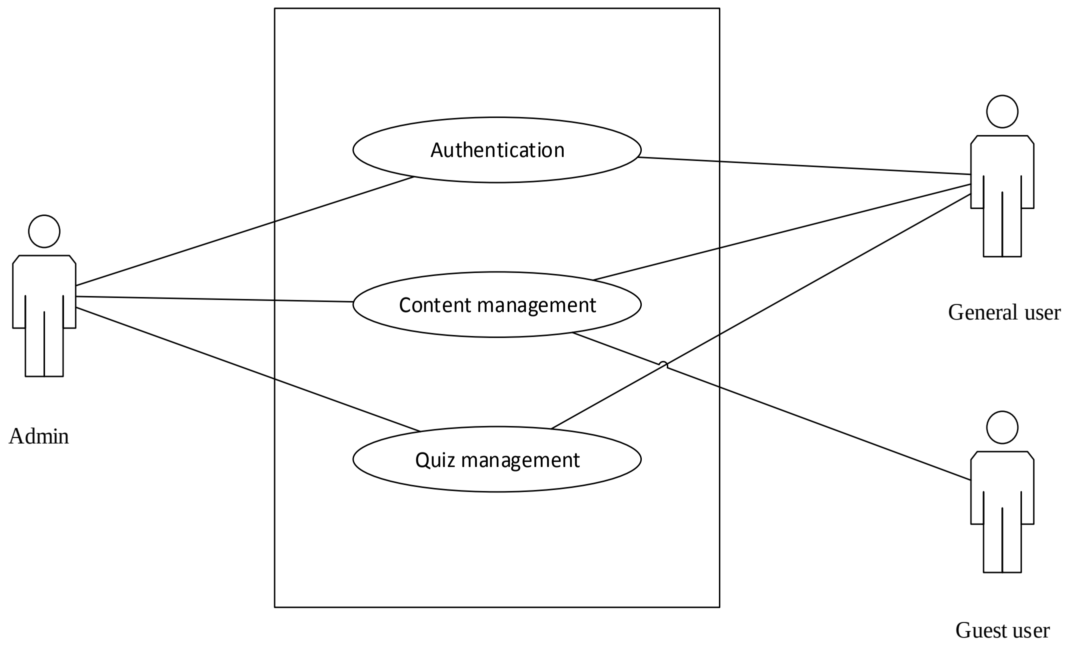


Figure 4. 1: Use case diagram (Level 1)

This is the level-1 of the use case for School of Programming. The actors involved in level-1 are general user, admin and guest user.

In figure 2 we have shown the relationship between all the actors and Authentication. We have broken the level 1 into some sub systems.

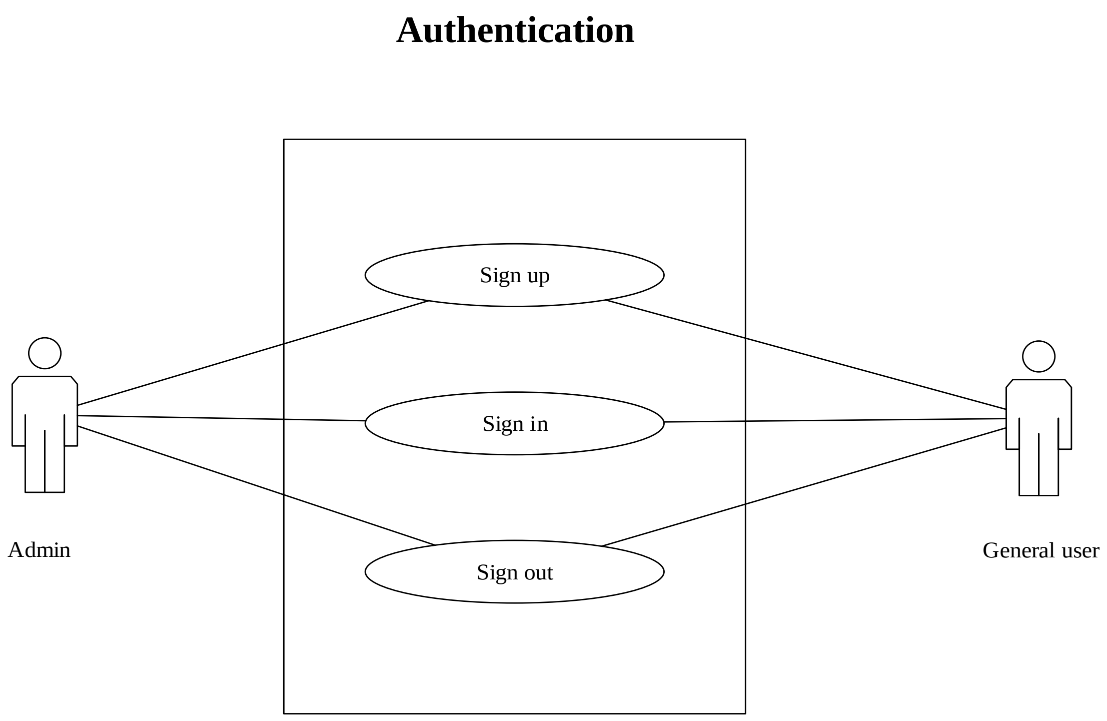


Figure 4. 2: Use case diagram (Level 1.1)

4.2.1 Authentication

Authentication system is divided into three sub-systems.

1.1 Sign Up

Use Case: Sign Up

Primary Actors: admin, general user

Goal in context: To register in the system

Precondition:

1. System has been programmed for add new user in database

2. System has interface for registration

Triggers: The user and admin need to register

Scenario:

1. Visit the register page
2. Input required information
3. Check availability for username & check validity of Password
4. E-mail sent to user e-mail address
5. Confirmation message showed

Exception:

- User is not authorized for registration
- Ambiguous Input
- Authentication Fail

Priority: Essential, must be implemented

When Available: First increment

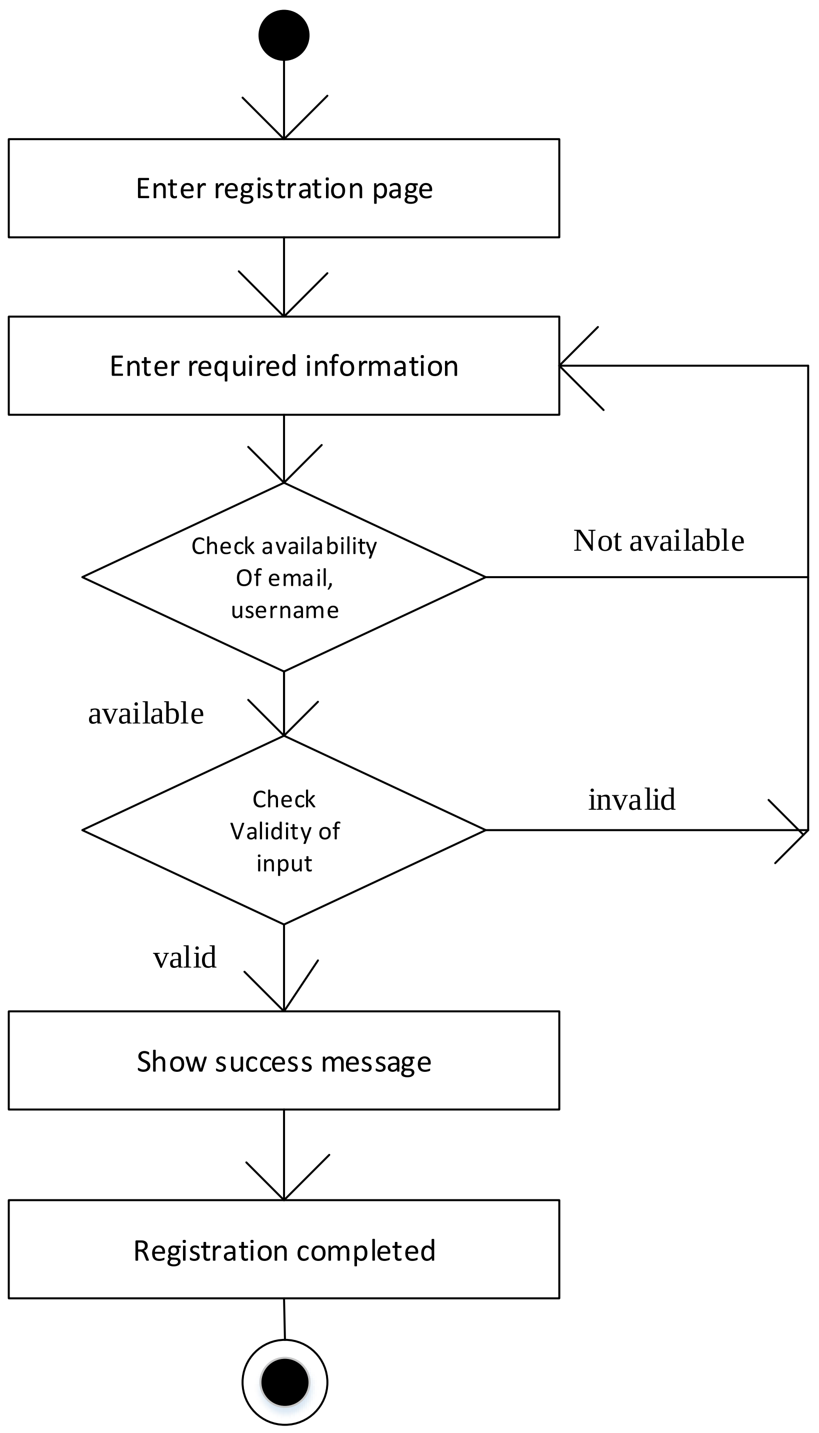


Figure 4. 3: Activity Diagram (Sign up)

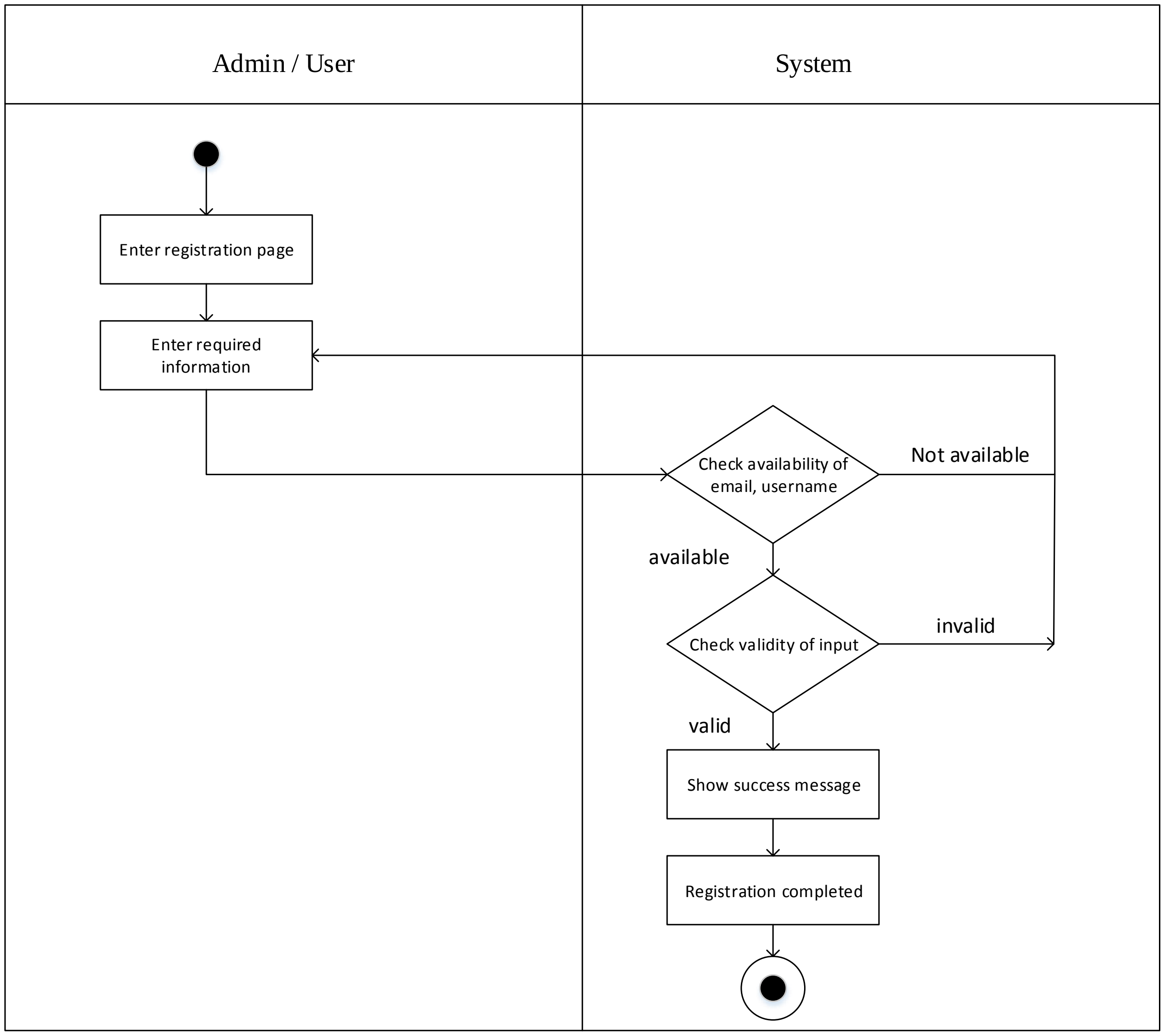


Figure 4. 4: Swim lane Diagram (Sign up)

1.2 Sign In

Use Case: Sign In

Primary Actors: Admin, general user

Goal in context: To enter the system

Precondition: Must be registered

Triggers: Need to login the system

Scenario:

1. Visit the login page

2. Input Username & Password

3. Proceed to the next activity

Exception:

- Unrecognized Username
- Wrong Password

Priority: Essential, must be implemented

When Available: First increment

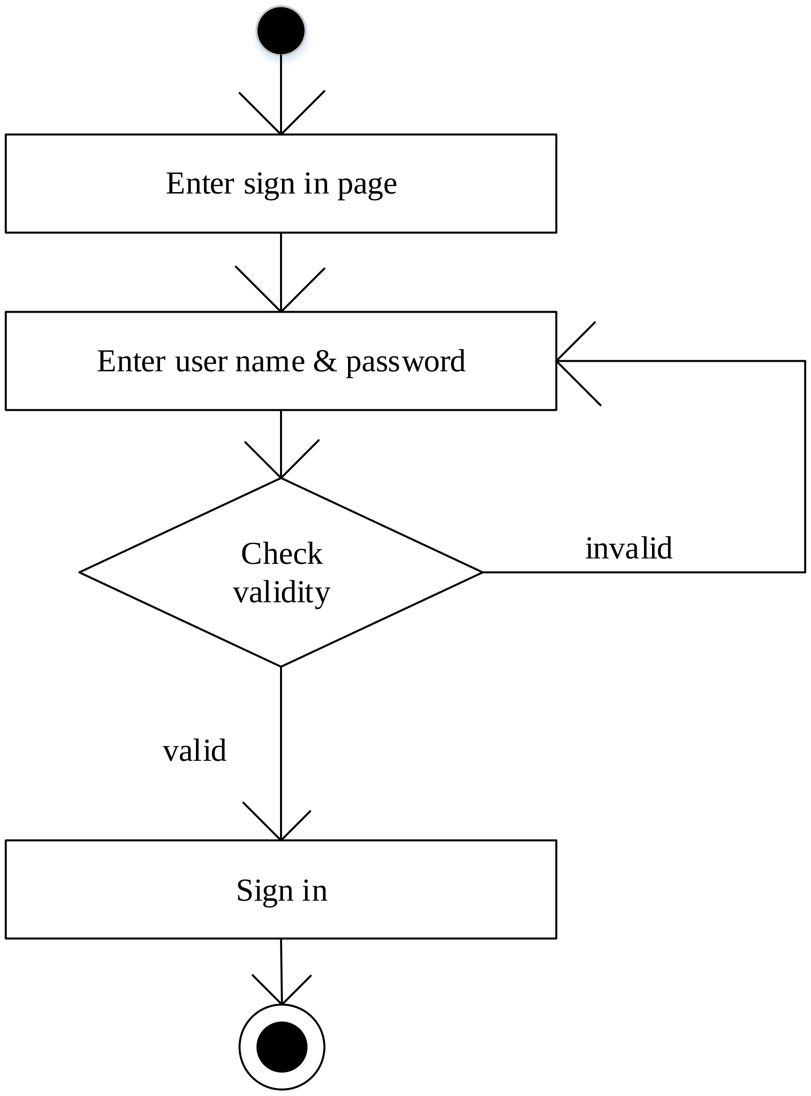


Figure 4. 5: Activity Diagram (Sign in)

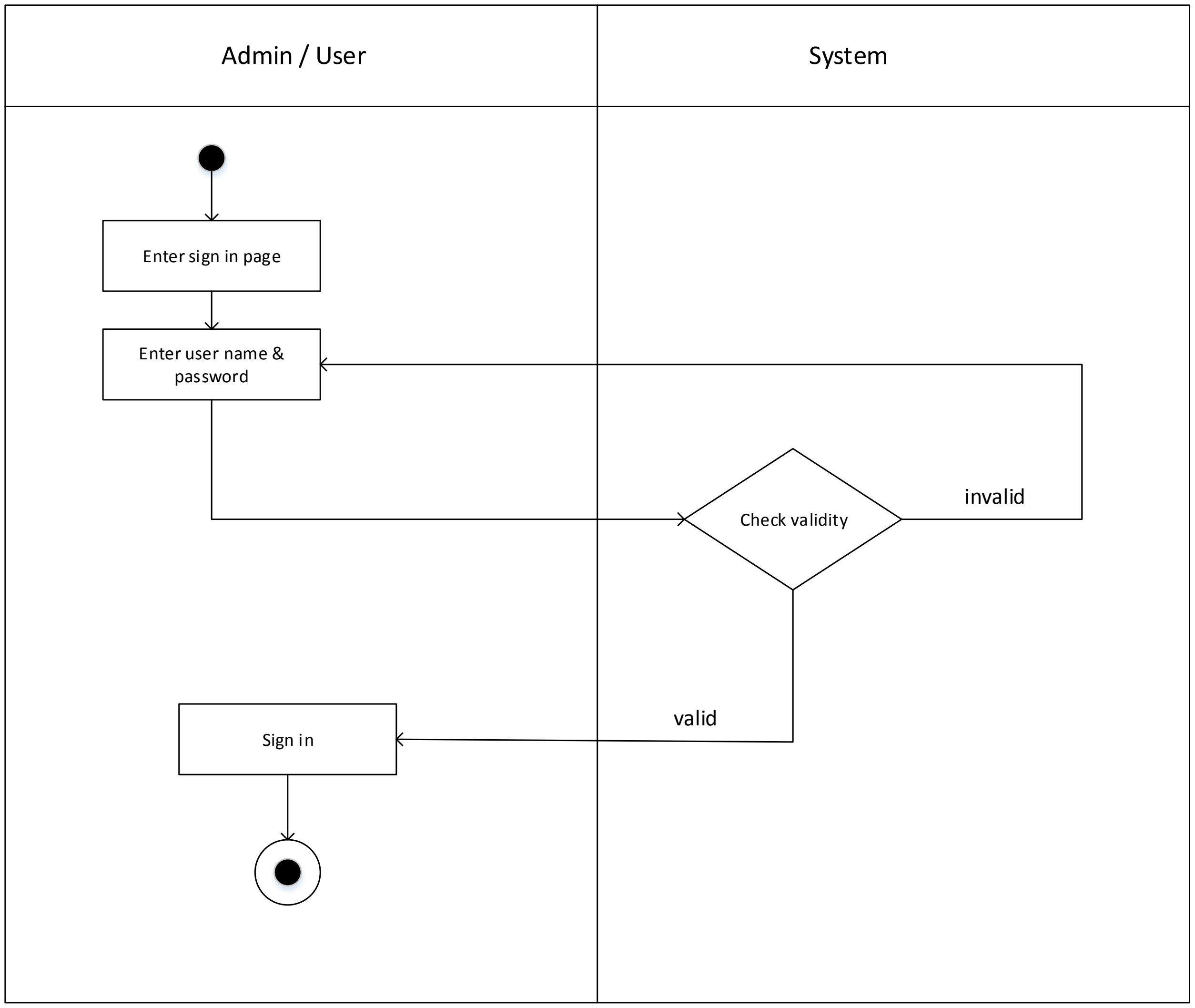


Figure 4. 6: Swim lane Diagram (Sign in)

1.3 Sign Out

Use Case: Sign Out

Primary Actors: Admin, general user

Goal in context: To exit from the system

Precondition: Must be logged in

Triggers: Need to log out from the system

Scenario: Click the logout button

Priority: Essential, must be implemented

When Available: First increment

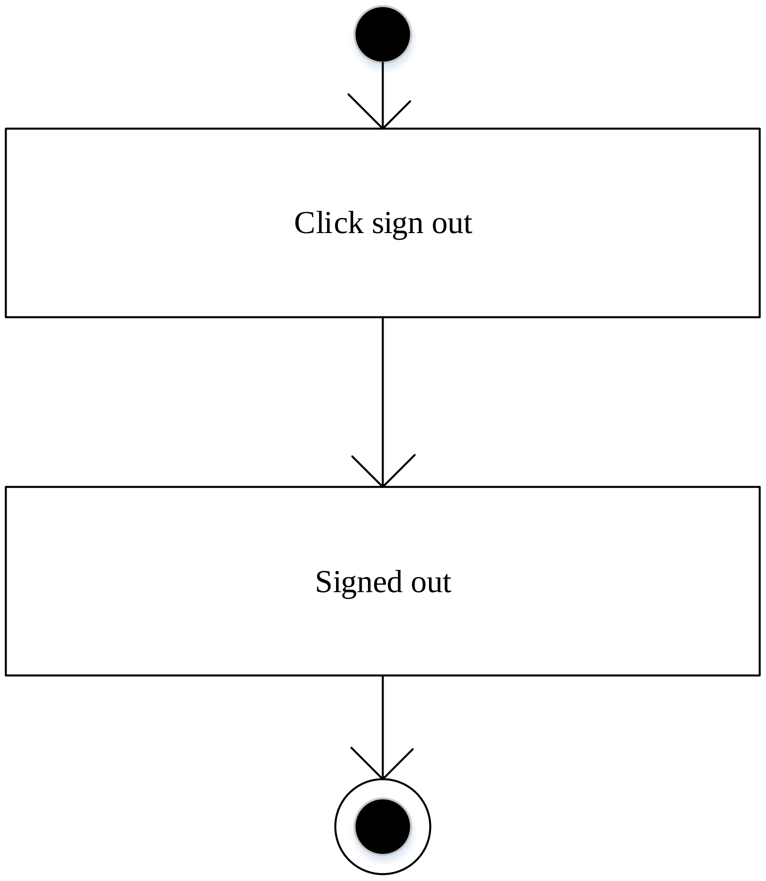


Figure 4. 7: Activity Diagram (Sign out)

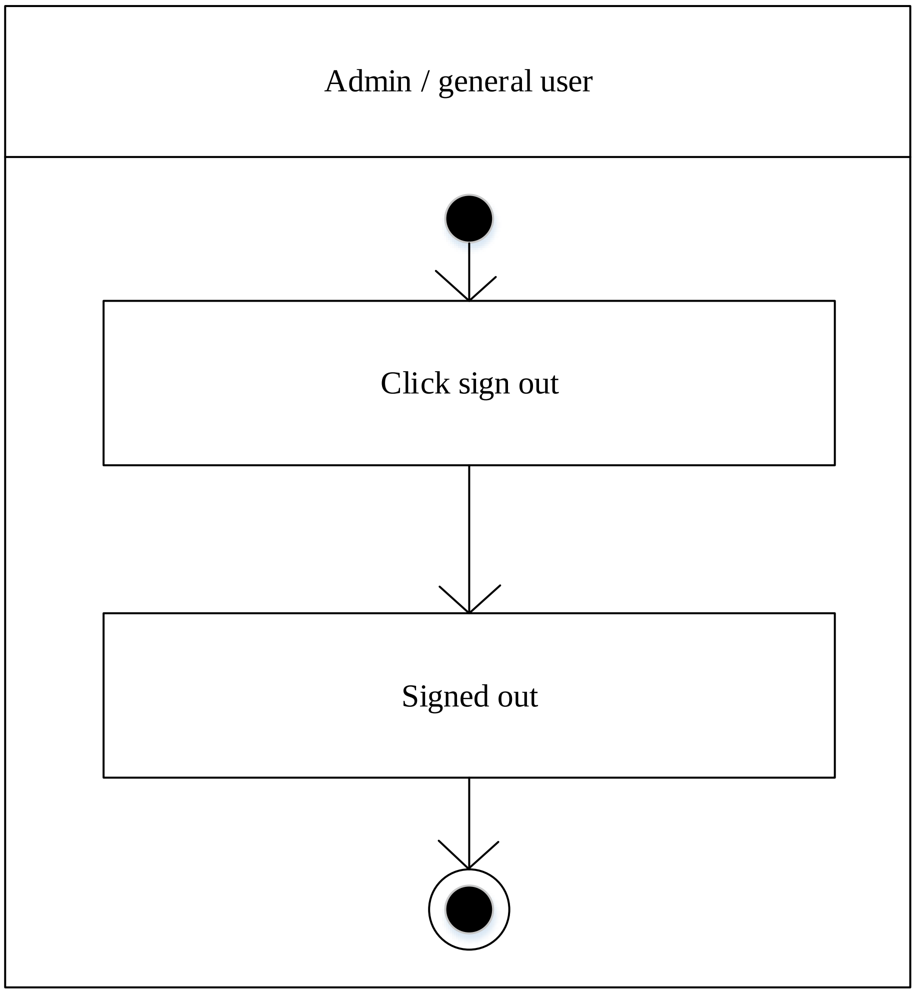


Figure 4. 8: Swim lane Diagram (Sign out)

In the following figure, we have shown the relationship between all the actors and Content management.

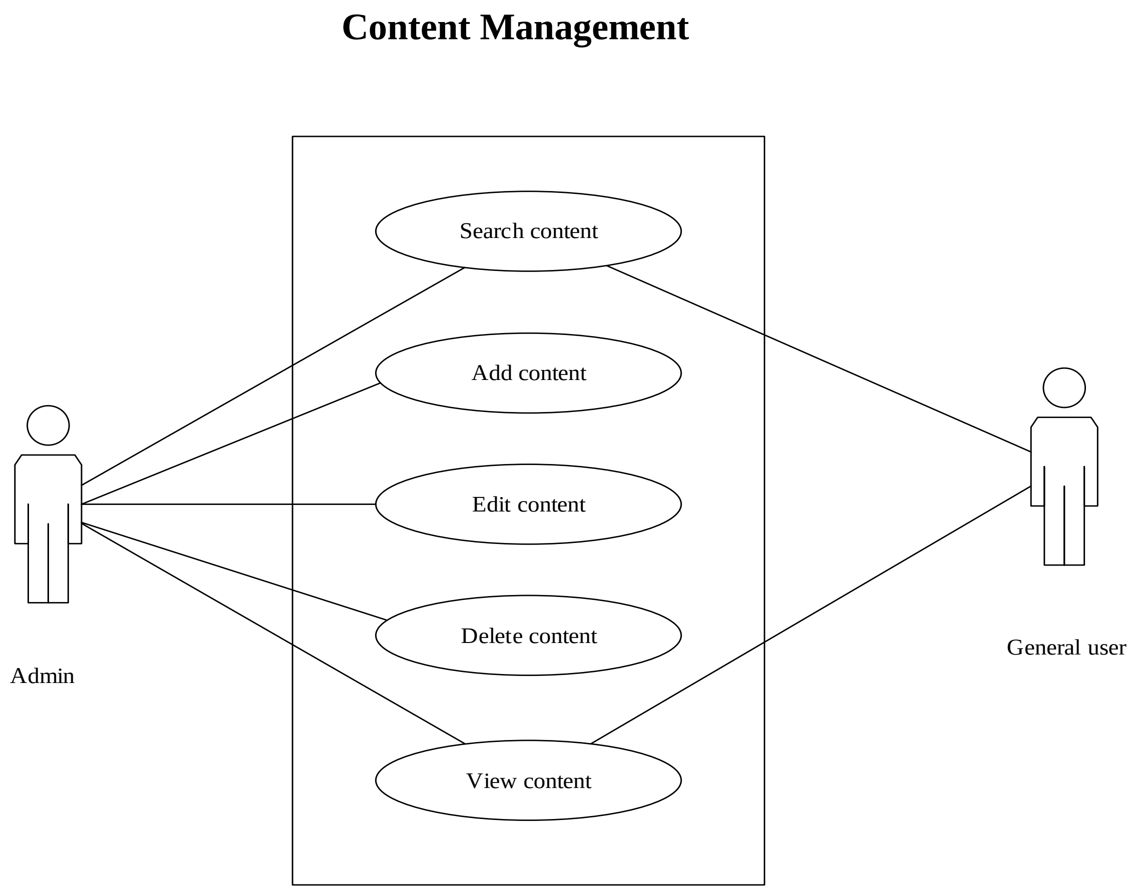


Figure 4. 9: Use case diagram (Level 1.2)

4.2.2 Content management

Content management is divided into five sub-systems.

2.1 Search content

Use Case: Search for content(s)

Primary Actors: Admin, user

Goal in context: To perform a search for content(s)

Precondition: System has been programmed for searching all contents in database

Triggers: The admin and users have need to search for item(s)

Scenario:

1. Enter data and information such as topic, chapter etc.
2. Click the Search button
3. View the search content
4. Proceed to the next activity

Exception:

- Search item does not exist

Priority: Essential, must be implemented

When Available: First increment

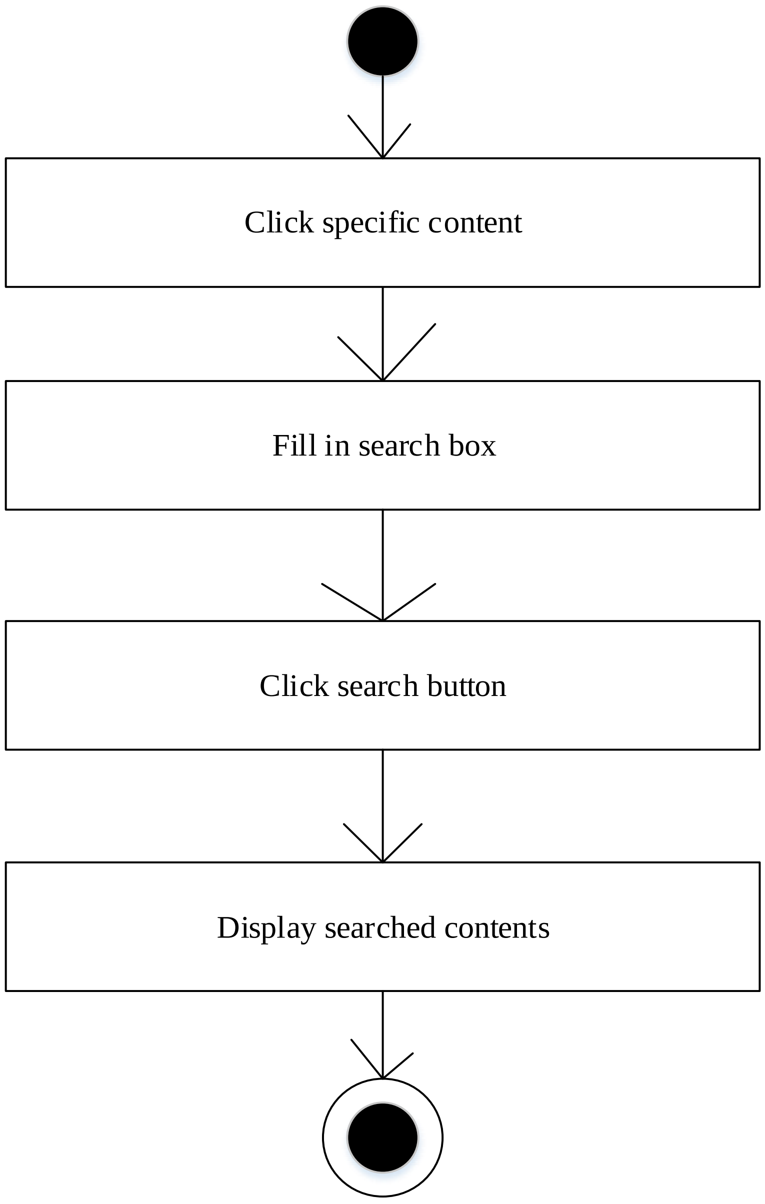


Figure 4. 10: Activity Diagram (Search content)

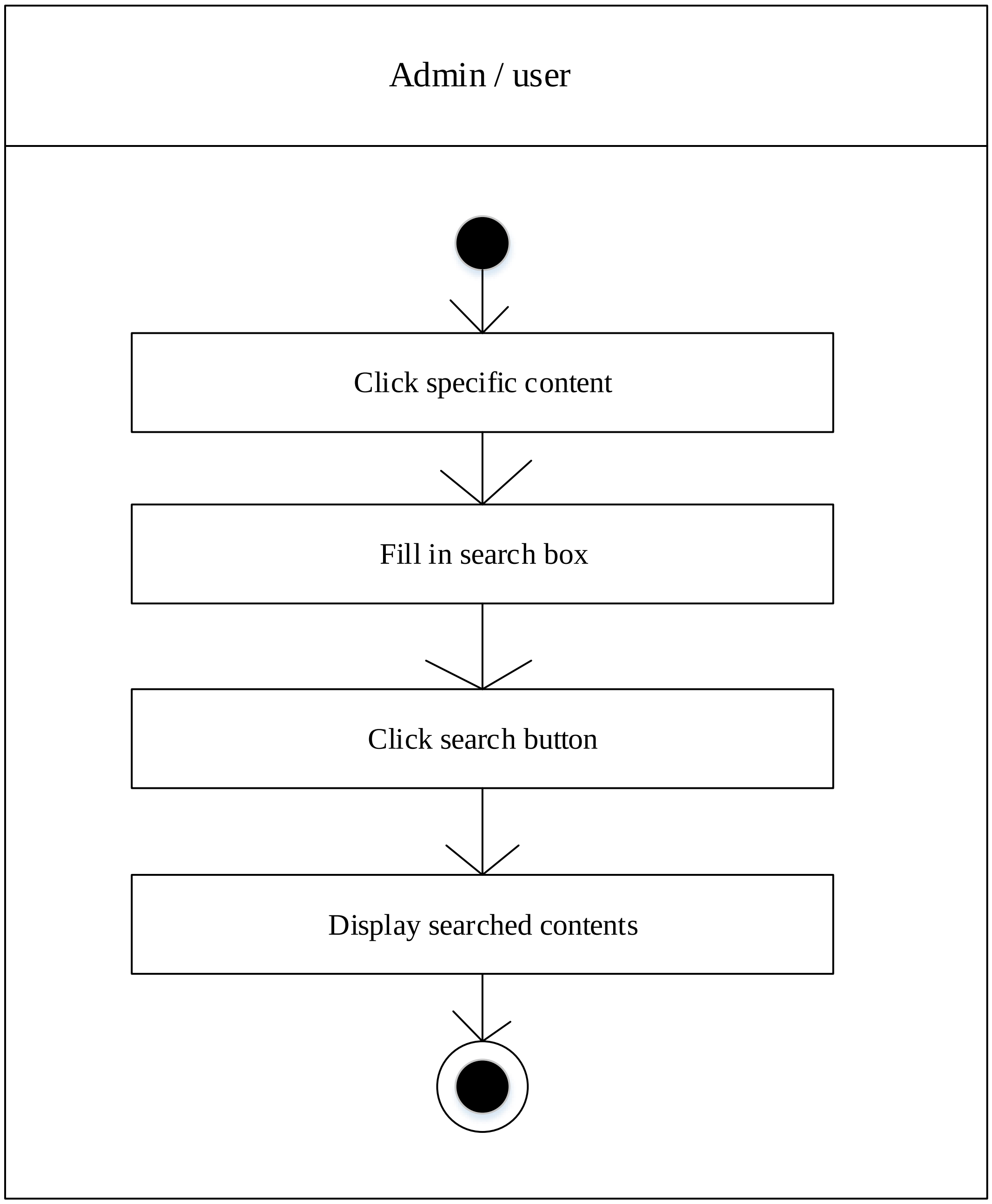


Figure 4. 11: Swim lane Diagram (Search content)

2.2 Add content

Use Case: Add a content(s)

Primary Actors: Admin

Goal in context: To add new item(s)

Precondition:

- System has been programmed for adding contents in database
- Must be signed in as admin

Trigger: The admin has a need to add new content(s)

Scenario:

- Visit sign in page and sign in
- Click on content button
- Click on Add content button to add new item
- Enter the new Item data (select Location) and confirm changes
- Proceed to the next activity

Exception: Already exist

Priority: Essential, must be implemented

When Available: First increment

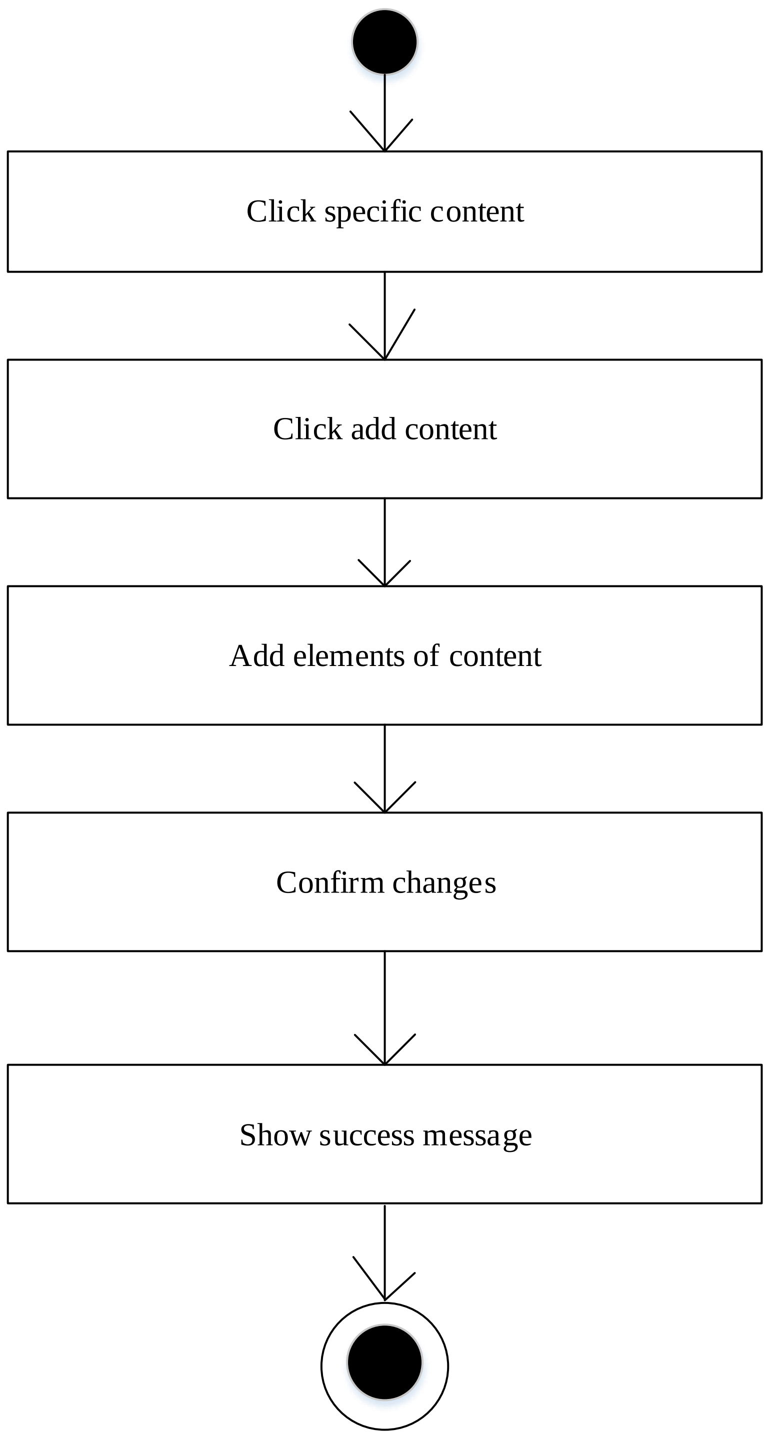


Figure 4. 12: Activity Diagram (Add content)

The swim lane diagram of add content is same as the swim lane diagram of search content with the single actor: admin.

2.3 Edit content

Use Case: Edit an Item

Primary Actors: admin

Goal in context: To edit a content

Precondition:

- System has been programmed for editing content in database
- Must be signed in as admin

Trigger: The admin has a need to edit a content(s).

Scenario:

1. Visit sign in page and sign in
2. Click on content button
3. Search and select the content to edit
4. Click on Edit Item button
5. Edit the Item details and confirm changes
6. Proceed to the next activity

Exception:

- Does not exist: Requested item does not exist in the database
- Wrong input

Priority: Essential, must be implemented

When Available: First increment

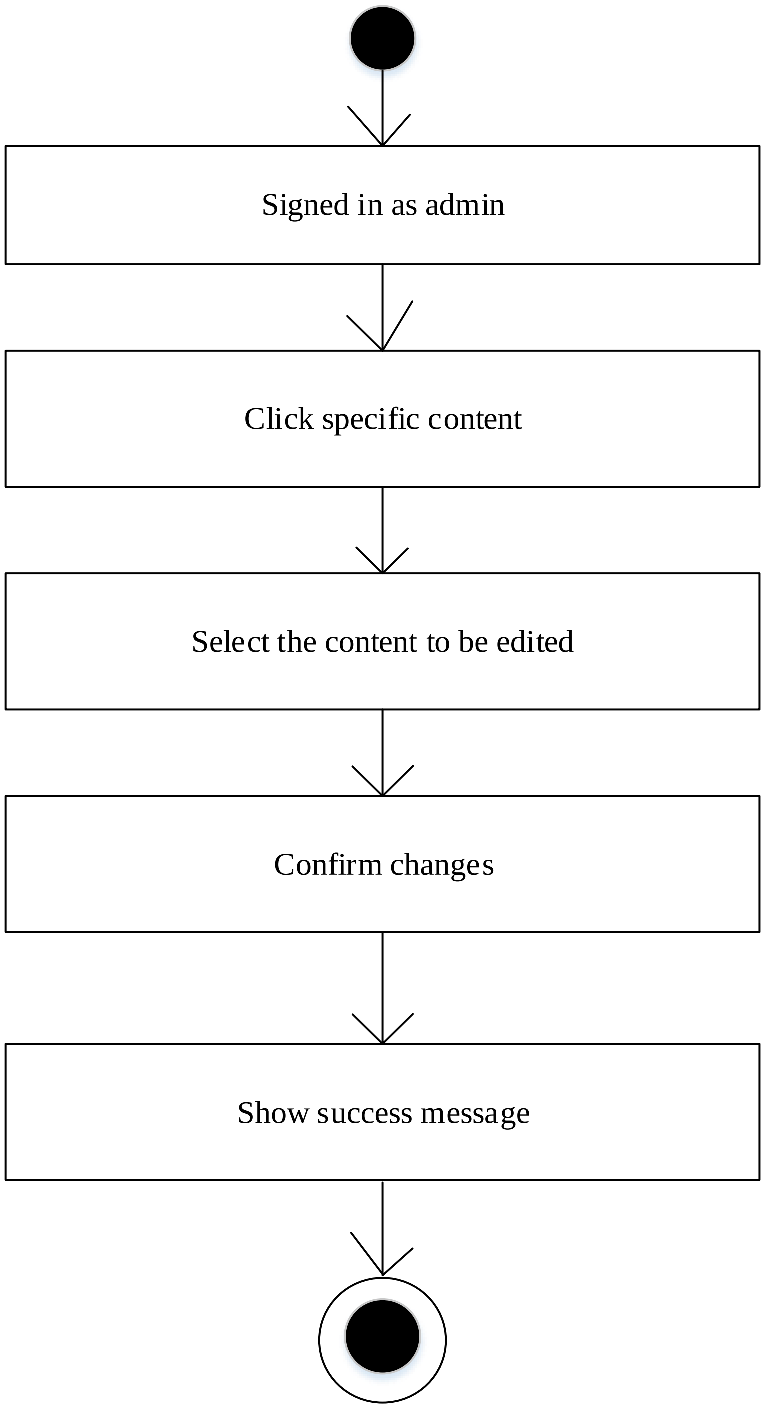


Figure 4. 13: Activity Diagram (Edit content)

The swim lane diagram of edit content is same as the swim lane diagram of search content with the single actor: admin.

2.4 Delete content

Use Case: Delete a content(s)

Primary Actors: admin

Goal in context: To delete a content

Precondition:

- System has been programmed for deleting content in database
- Must be signed in as admin

Trigger: The admin has a need to delete a content(s).

Scenario:

- Visit sign in page and sign in
- Click on content button
- Search and select the content to delete
- Click on delete Item button
- Delete the selected Item and confirm changes
- Proceed to the next activity

Exception: Item does not exist.

Priority: Essential, must be implemented

When Available: First increment

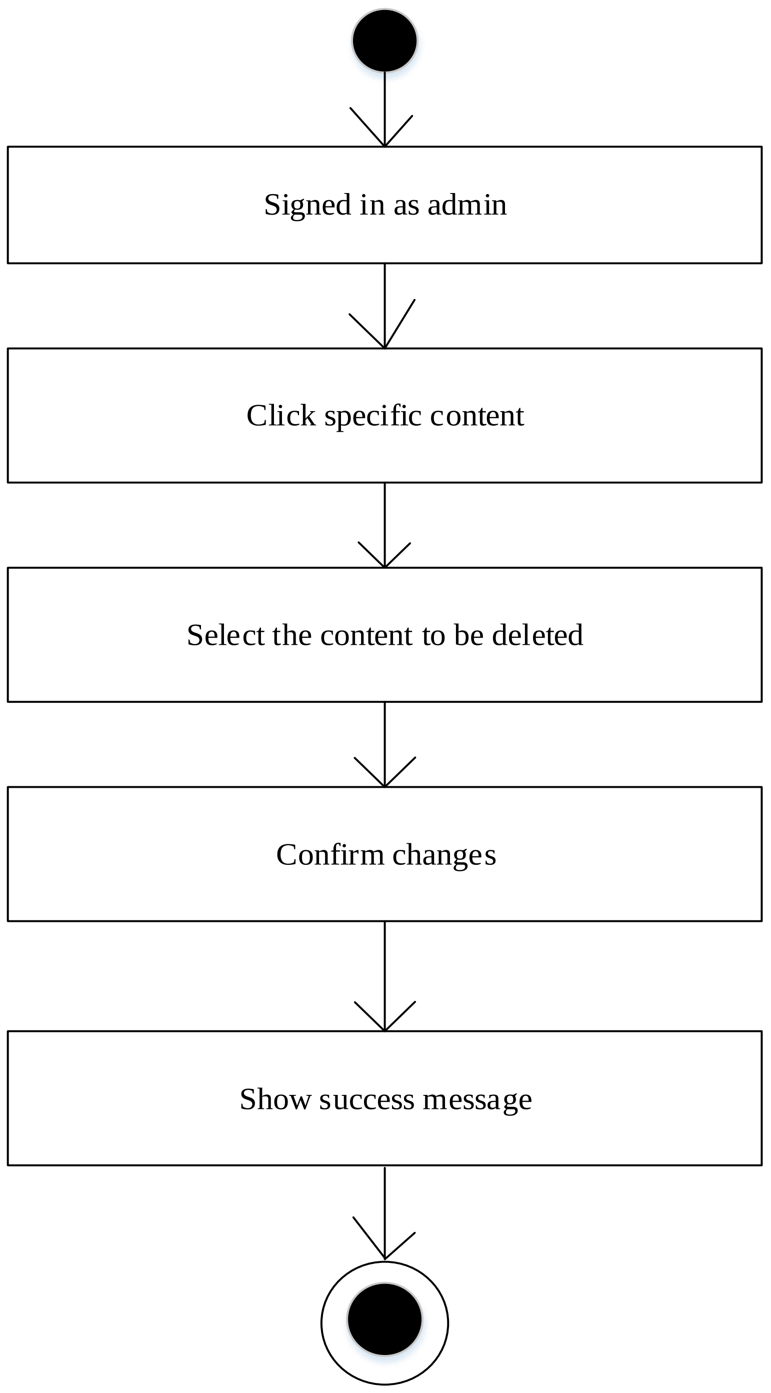


Figure 4. 14: Activity Diagram (Delete content)

The swim lane diagram of delete content is same as the swim lane diagram of search content with the single actor: admin.

2.5 View content

Use Case: View a content(s)

Primary Actors: Admin, general user, guest user

Goal in context: To perform a view for content(s)

Precondition: System has been programmed for searching all contents in database

Triggers: The admin and users have a need to view any item

Scenario:

- Enter data and information such as topic, chapter etc.
- View the search content
- Proceed to the next activity

Exception:

- Desirable item does not exist

Priority: Essential, must be implemented

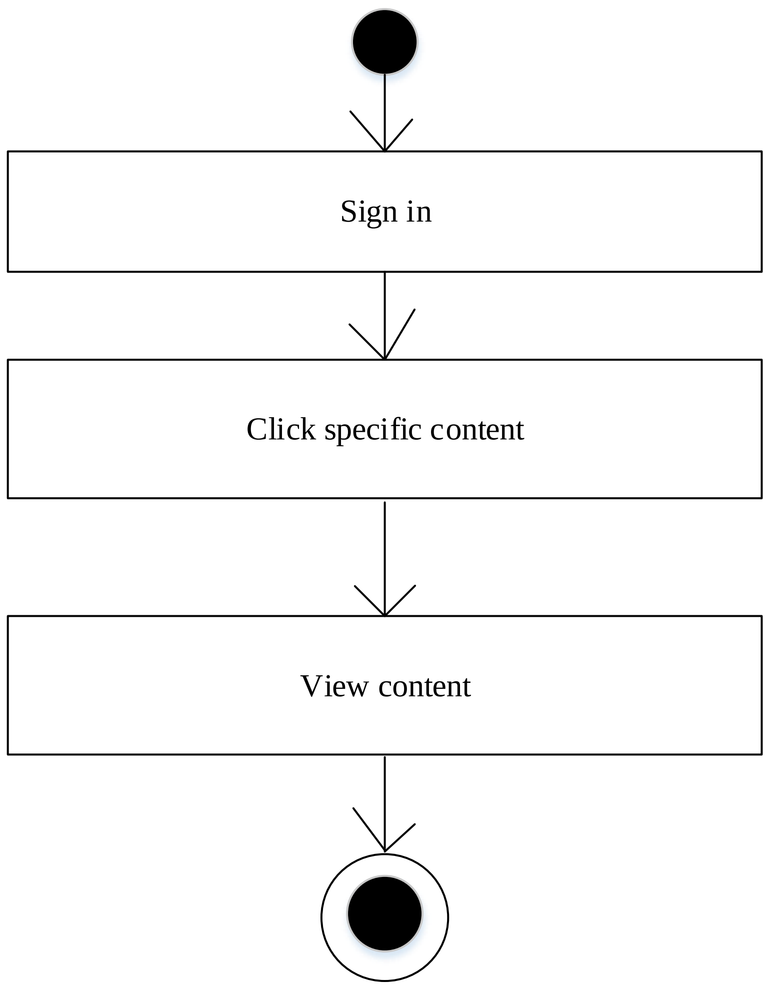


Figure 4. 15: Activity Diagram (View content)

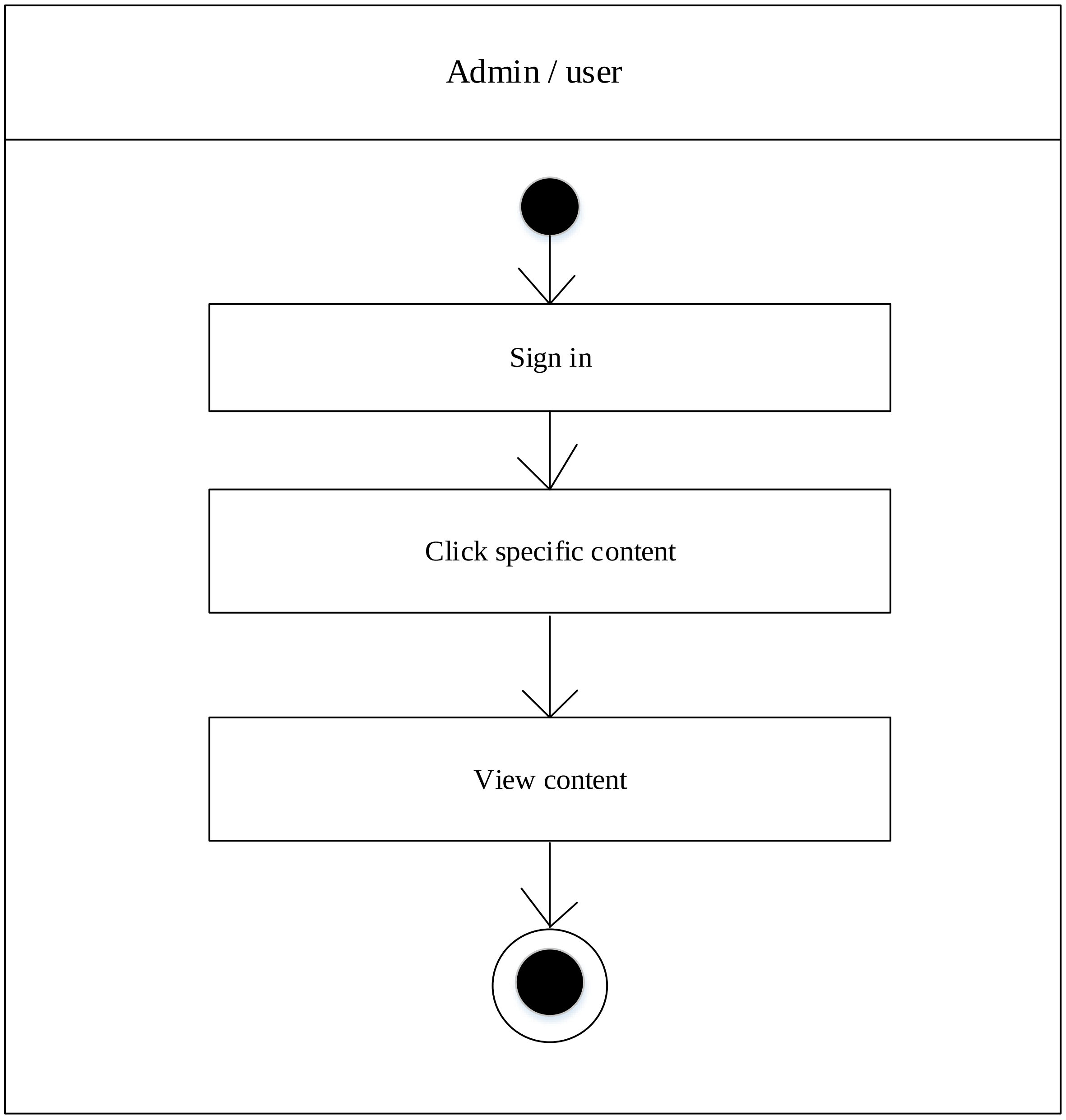


Figure 4. 16: Swim lane Diagram (View content)

In the following figure, we have shown the relationship between all actors and quiz management module.

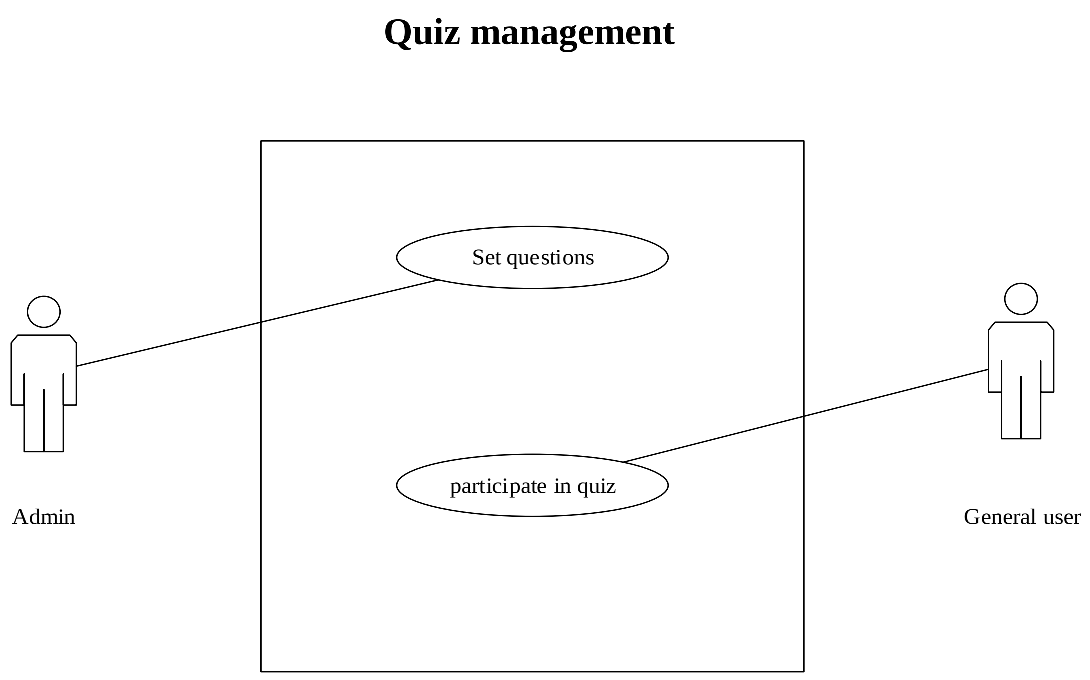


Figure 4. 17: Use case diagram (Level 1.3)

4.2.3 Quiz management

3.1 Set questions

Use Case: set questions

Primary Actors: admin

Goal in context: Set questions for taking quiz

Precondition:
- Must be signed in
- System has interface for adding question

Triggers: The admin needs to add questions for taking quiz

Scenario:

- Visit the quiz page
- Add the question and submit
- Proceed to the next activity

Exception: Wrong format

Priority: Essential, must be implemented

When Available: First increment

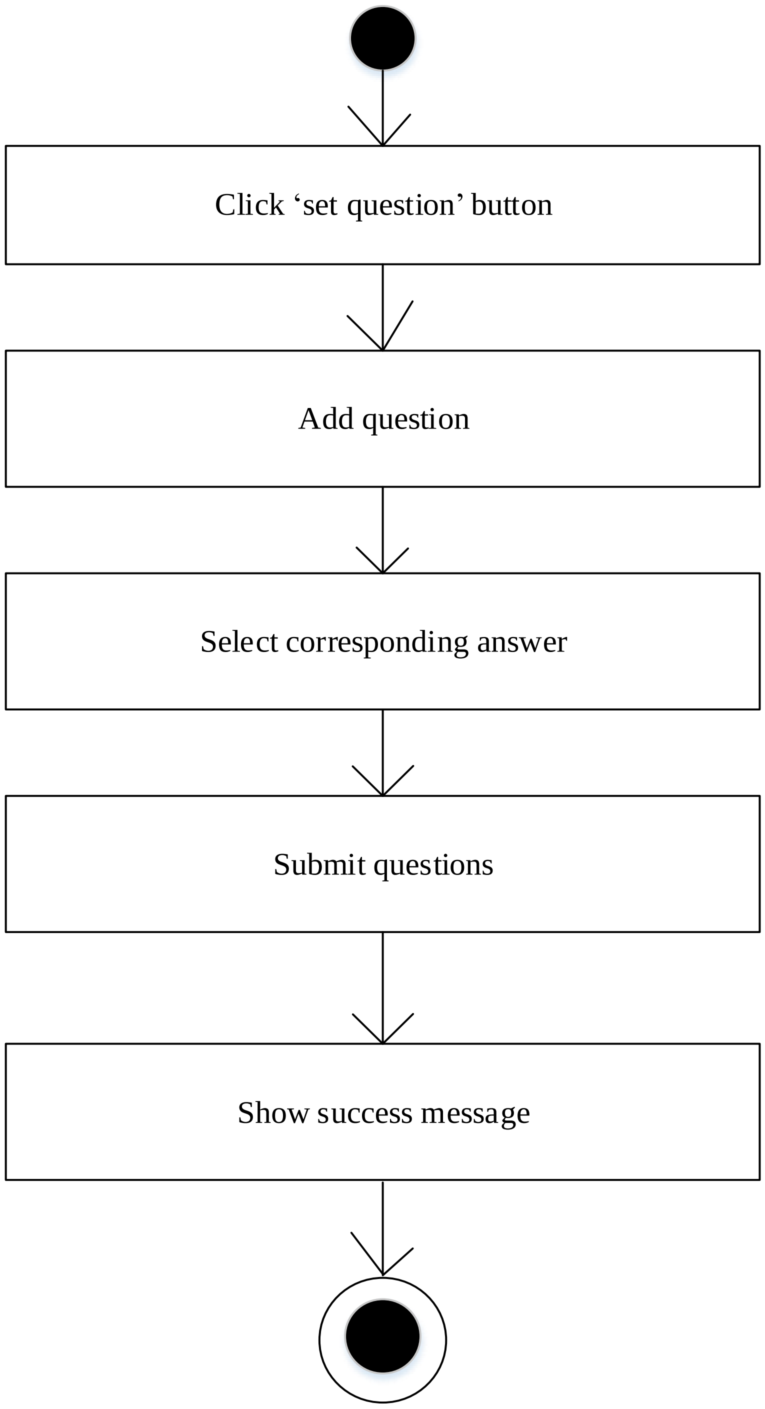


Figure 4. 18: Activity Diagram (Set questions)

The swim lane diagram of set questions is same as the activity diagram of set questions with the single actor: admin

3.2 Participate in quiz

Use Case: participate in quiz

Primary Actors: general user

Goal in context: Reply answer of a question
Precondition:

- Questions must be added before answering
- System has interface for choosing answers

Triggers: The user needs to answer the question

Scenario:

- Visit the quiz page
- Select the answers and submit
- Proceed to the next activity

Exception: User is not authorized for participating in the quiz

Priority: Essential, must be implemented

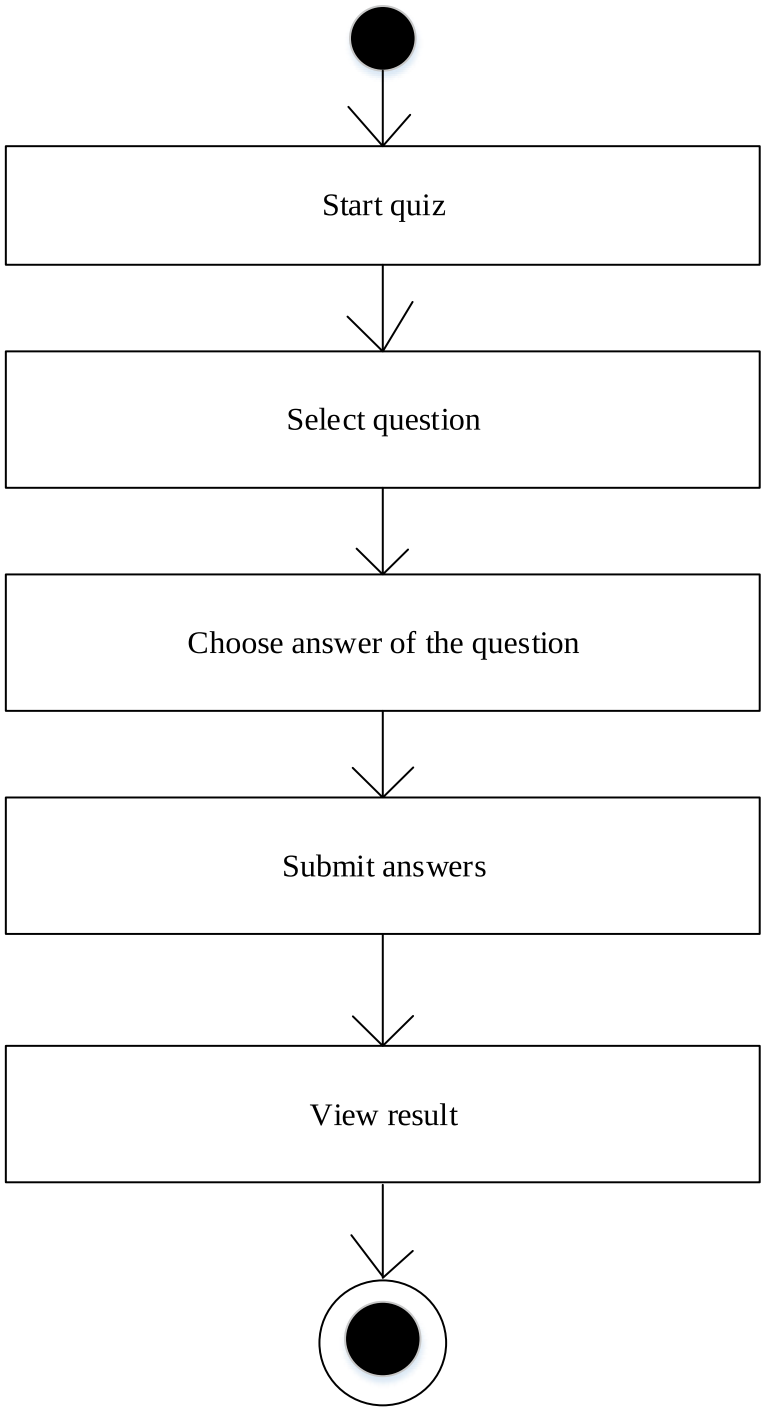


Figure 4. 19: Activity Diagram (Participate in quiz)

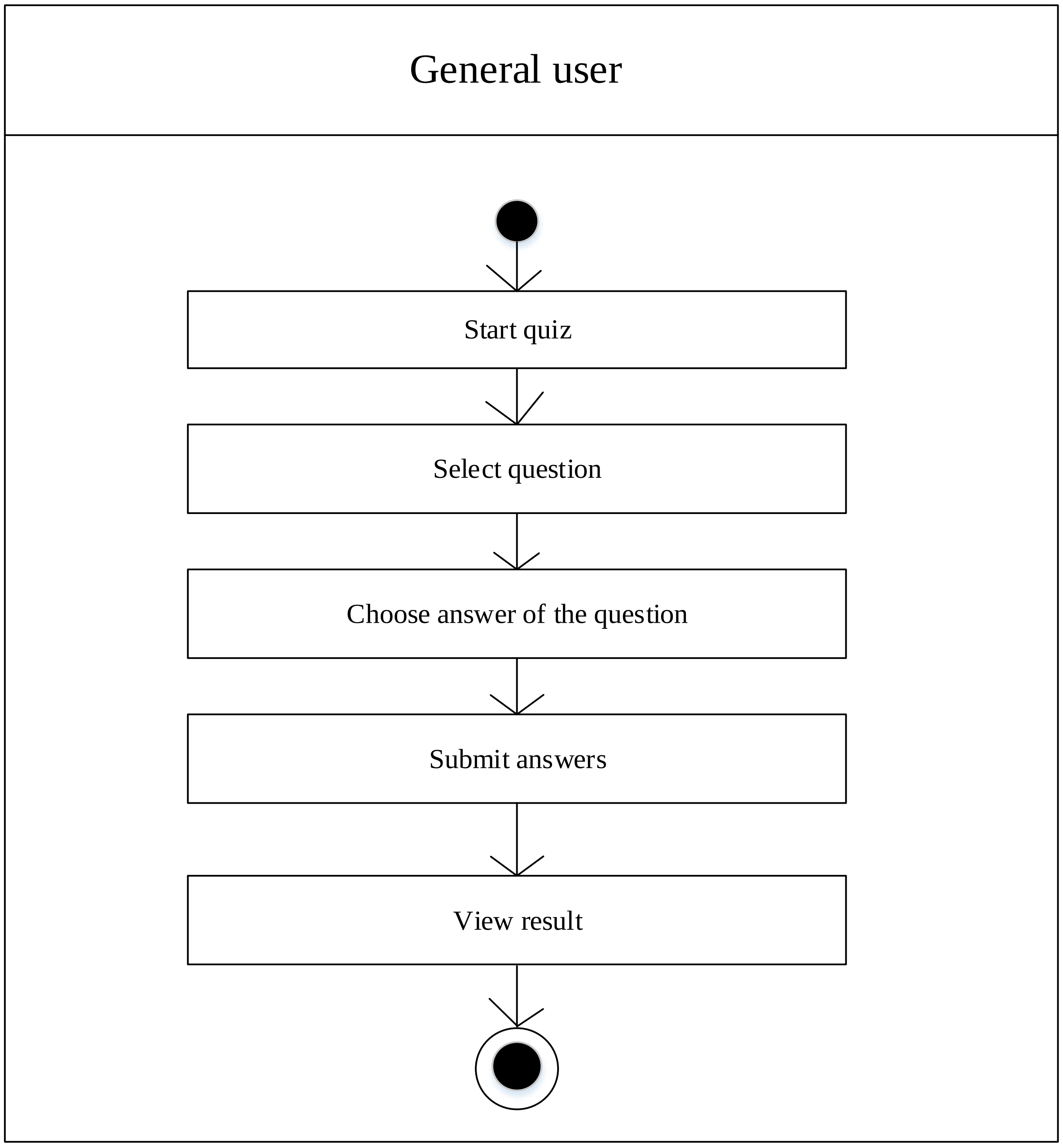


Figure 4. 20: Swim lane Diagram (Participate in quiz)

Chapter 5

Data Modelling

5.1 Data modelling basics

Data modeling is the process of determining data requirements from business requirements. The data requirements are captured in an entity-relationship diagram, also known as E-R diagram. The E-R diagram contains entities and their relationships. As the model evolves, attributes are identified for the entities. Normalization is utilized to determine the identifying and dependent attributes within the entities and to remove many-to-many relationships between the entities. Once normalized the entities and attributes are converted to tables and columns to become a relational database.

5.2 Data objects

Inferring the data objects identifies how the data within a table or file relates to the data in other tables or files. Understanding how the data within a table or file relates to other tables and files is important to realize the business relationships contained within a data model. Data models that contain hundreds or thousands of tables or files make it difficult to locate and understand the business relationships between the tables or files. Profiling these complicated models groups the tables and files together that contain similar or related data. This simplifies and accelerates the entire modeling process.

The data objects of this scenario are as following:

Data Object: User

Attributes:

- User_id
- Name
- Email
- Password

Data Object: Admin

Attributes:

- Admin_id
- Password
- Email
- Name

Data Object: Content

Attributes:

- Content_id
- Topic
- Details
- Name

Data Object: Quiz

Attributes:

- Quiz_id
- Question
- Answer
- Option
- Date
- Marks

Data Object: Topic

Attributes:

- Topic_id
- Topic_name

5.3 E-R Diagram

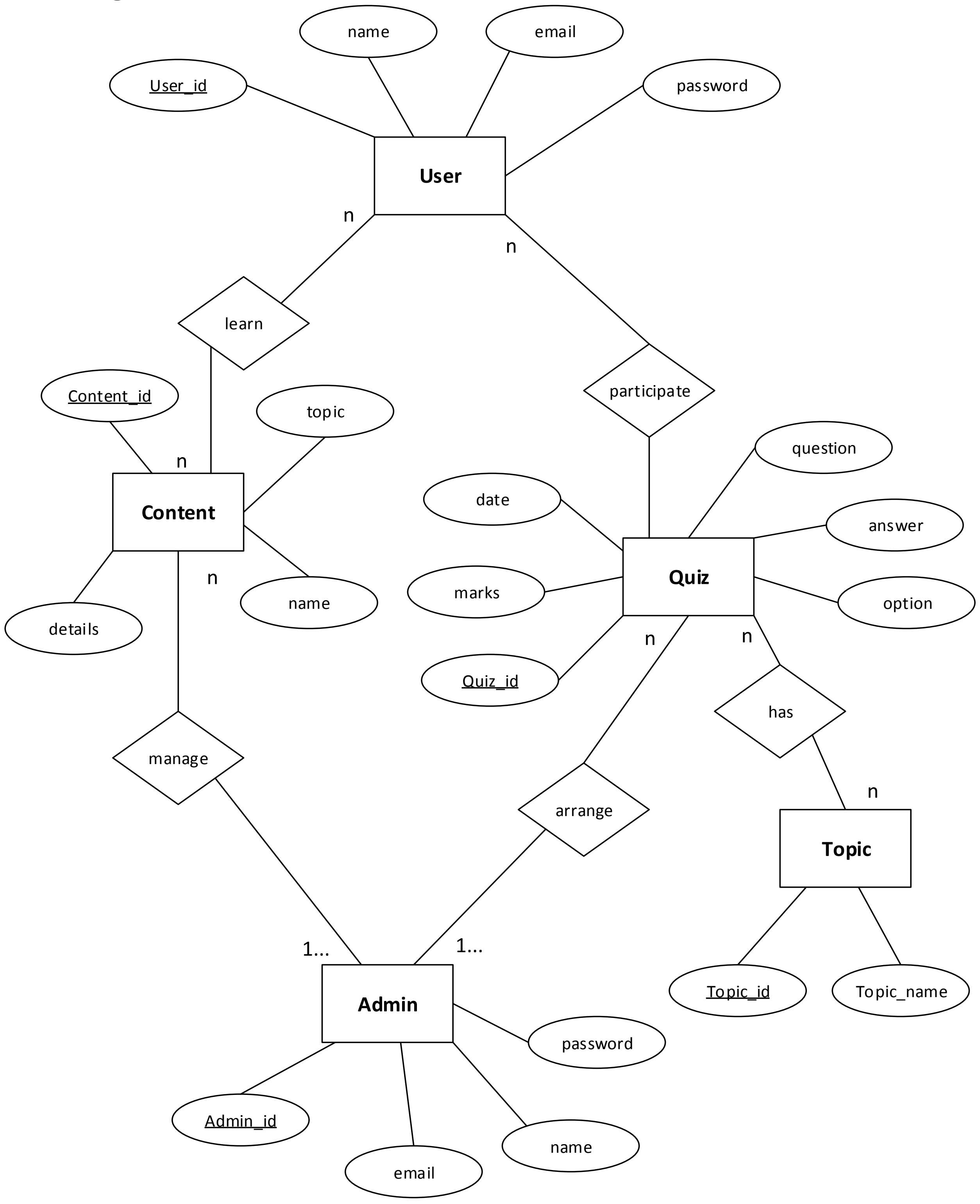


Figure 5. 1: E-R Diagram

Chapter 6

Class based modelling

The class based modelling is described in this part of the document.

6.1 Class Based Modelling Concepts

Class-based modeling represents the objects that the system will manipulate, the operations that will applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.2 Identifying analysis classes

- Identify analysis classes by examining the problem statement
- Use a “grammatical parse” to isolate potential classes
- Identify the attributes of each class
- Identify operations that manipulate the attributes.

Analysis Classes manifest themselves in one of the following ways:

1. **External entities** (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
2. **Things** (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. **Occurrences or events** (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. **Roles** (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. **Organizational units** (e.g., division, group, and team) that are relevant to an application.
6. **Places** (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. **Structures** (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

Performing a “grammatical parse” on a processing narrative for a problem helps extracting the nouns. After identifying the nouns, a number of potential classes are proposed in a list. The list will be continued until all nouns in the processing narratives have been considered. Each entry is in the list is a potential object.

1. **Retained Information:** The potential class will be useful during analysis only if information about it must be remembered so that the system can function.

2. **Needed Services:** The potential class must have a set of identifiable operations that can change the value of its attributes in some way.

3. **Multiple attributes:** During R.A., the focus should be on “major” information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.

4. **Common attributes:** a set of attributes can be defined for the potential class, and these attributes apply to all instances of the class.

5. **Common operations:** a set of operations can be defined for the potential class, and these operations apply to all instances of the class.

6. **Essential Requirements:** External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirement model.

6.3 Potential Classes:

To be considered a legitimate class for inclusion in the requirements model, the potential classes which satisfy all of these characteristics are following:

- Admin
- User
- Content
- Quiz
- Topic

6.4 Class Diagram

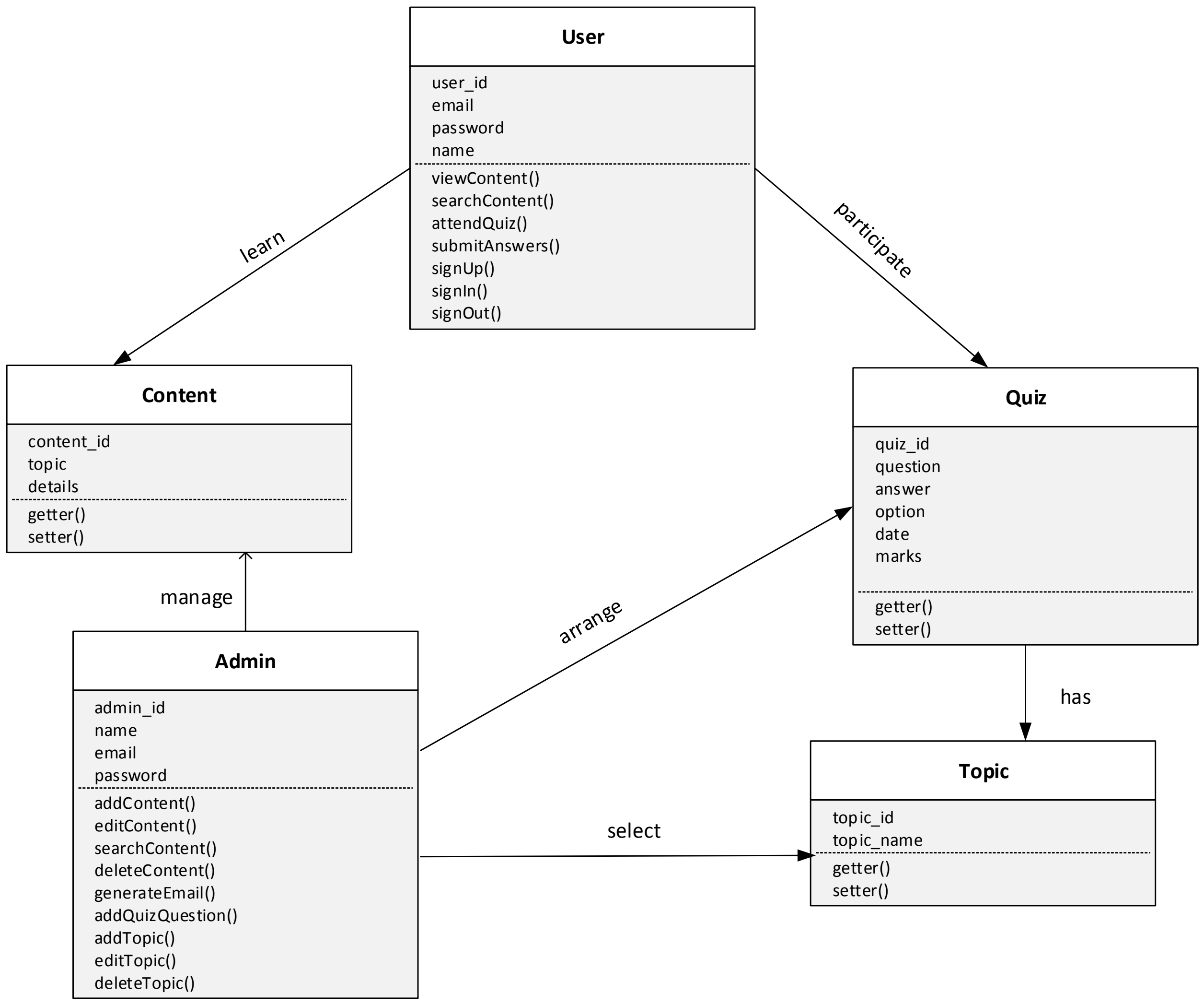


Figure 6. 1: Class diagram

Chapter 7

Behavioral Modelling

In this part of the document the brief description of behavioral model is given.

7.1 State Diagram

The following figures are state diagrams for the system. Each rectangle with rounded corners represents a different state of the system. The lines with arrows represent a transition with the arrow pointing to the new state. A circle with an arrow points to the beginning state. Conditions to take a given transition are given in brackets.

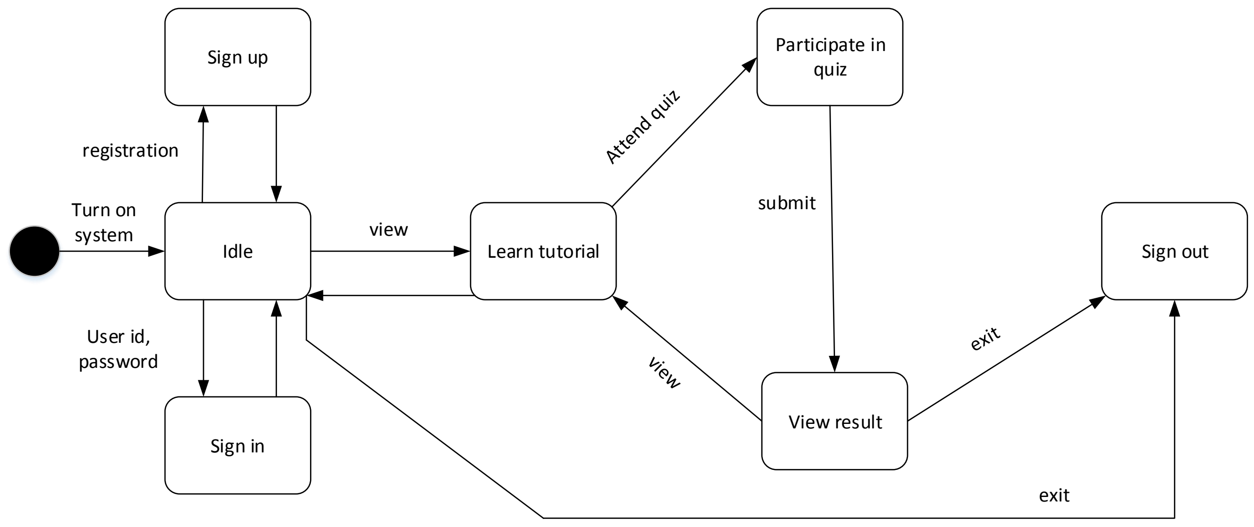


Figure 7. 1: State diagram of User class

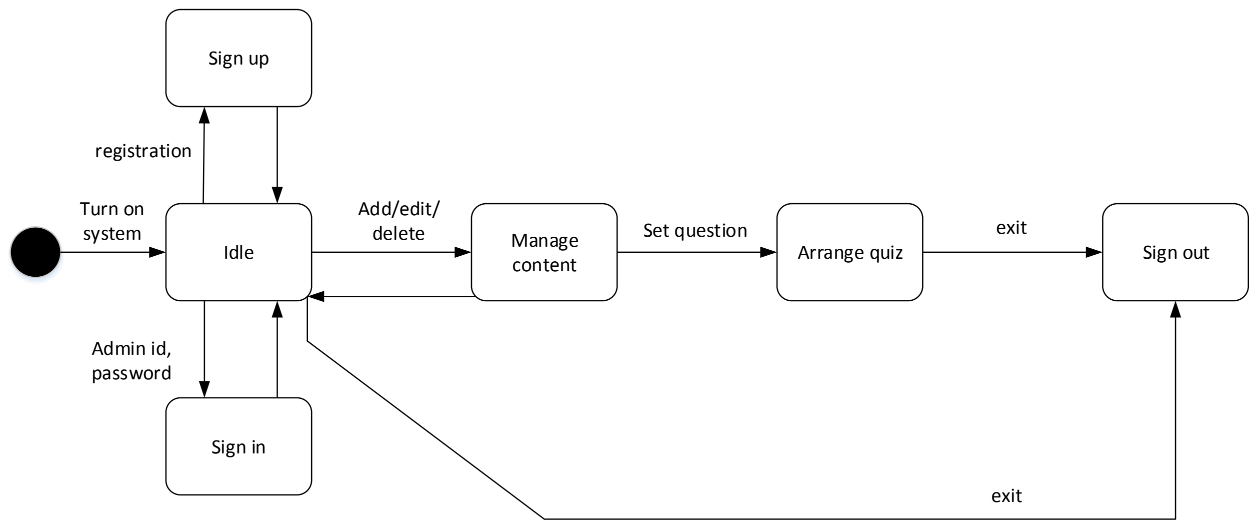


Figure 7. 2: State diagram of Admin class

7.2 Sequence diagram:

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

Sequence diagrams show specific scenarios that the system will go through. The boxes on the top represent different menus and pages while the distance from the top represents the length of time. The lines connecting the menus and pages represent transitions that are being sent at that particular point in time.

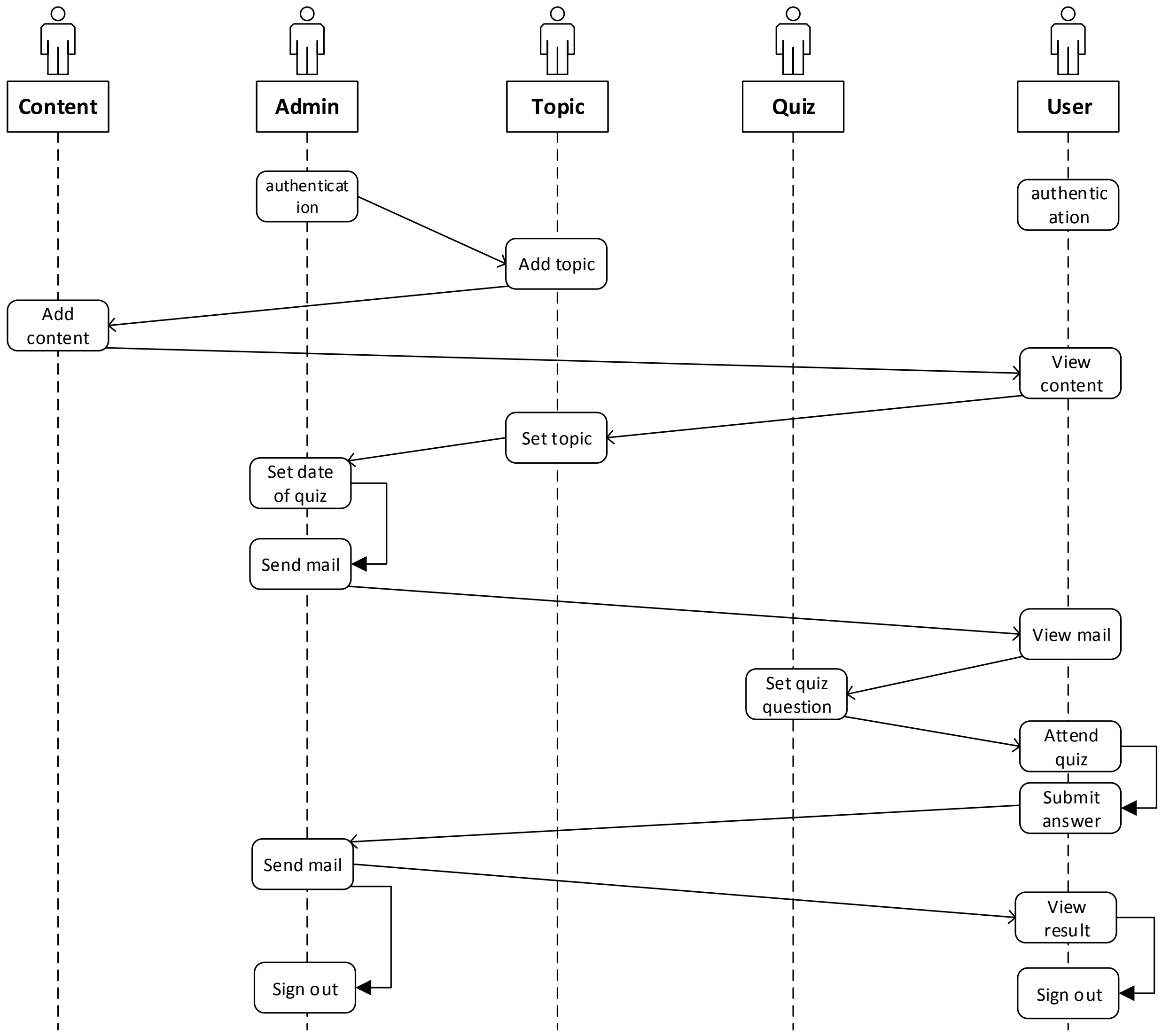


Figure 7. 3: Sequence diagram

Chapter 8

Software Design Architecture

8.1 Introduction

Software architectural design has been followed in “School of Programming”. At the beginning of the architectural design, the context diagram of the “School of Programming” is defined. Then the archetype of the project is described. And after the finding the archetypes, the components and the classes are defined.

8.2 Representing the system in context

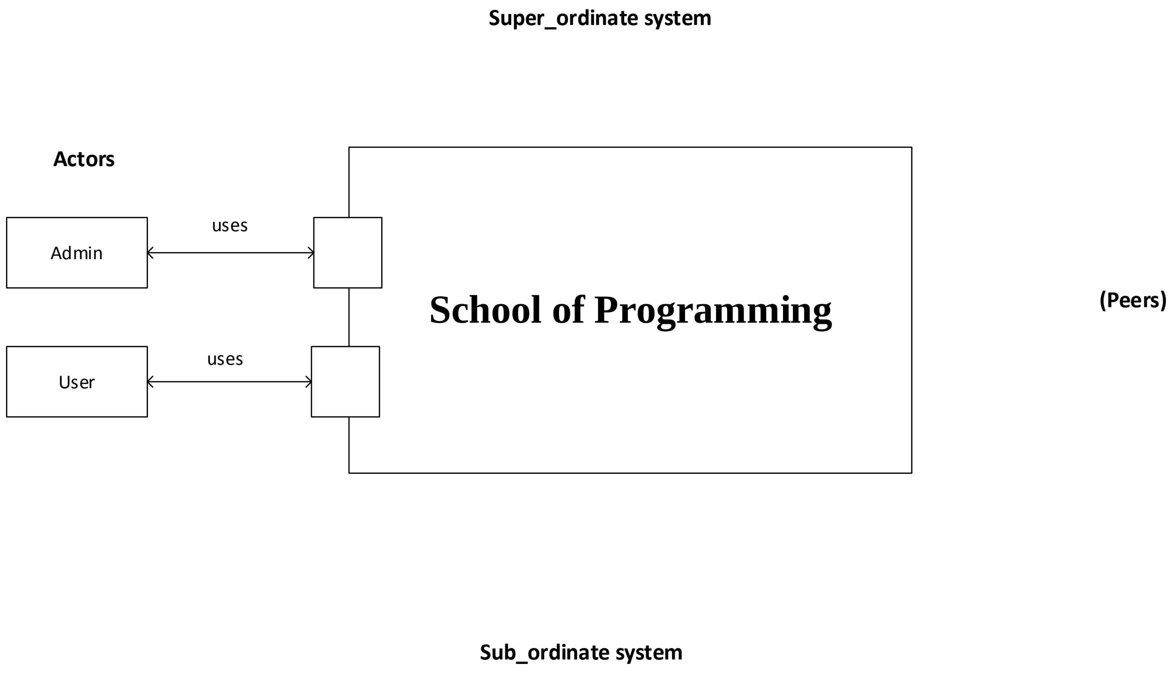


Figure 8. 1: Context Diagram

In the context diagram, the total system has two actors and they are: Admin and User. They are also the primary actor of this system. These actors give input to the system and receive the output from the system. “School of Programming” does not use any external subsystem.

8.3 Define archetypes:

1. Authentication
2. Content management
3. Quiz management

8.4 Refining archetypes into components

Components:

1. Sign up, sign in, Sign out, DAL
2. Add content, edit content, delete content, search content, view content, DAL
3. Set quiz questions, participate quiz, set topic, DAL

Classes:

1. User, admin
2. Admin, user, content
3. Admin, user, quiz, topic

8.5 Describing instantiation of the system

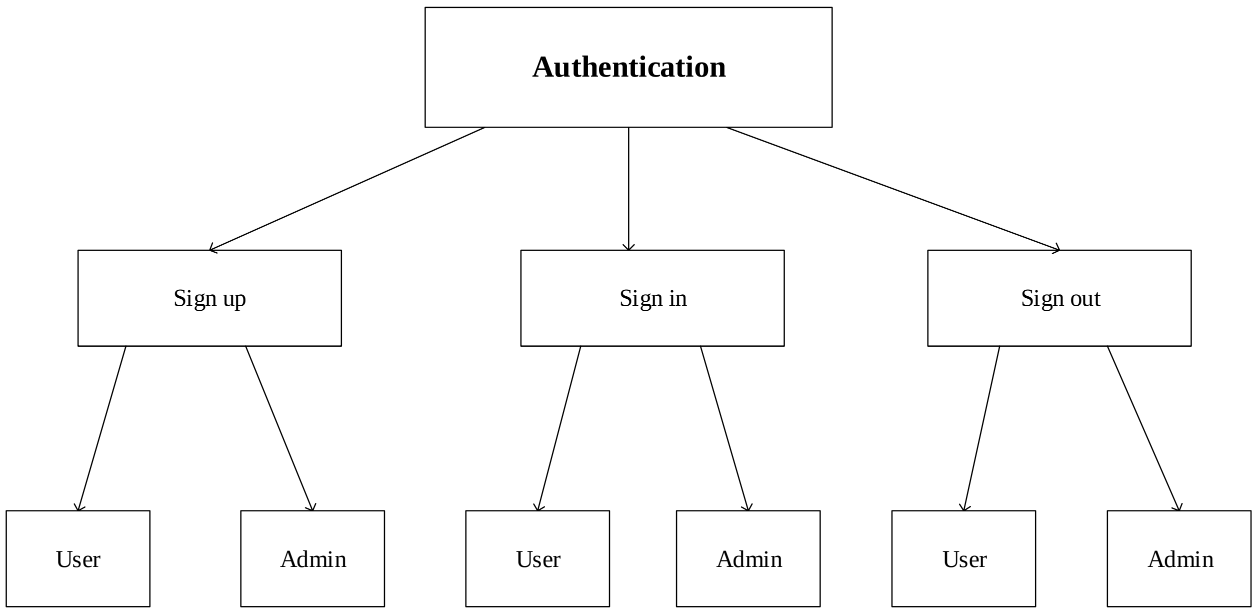


Figure 8. 2: Refining 'Authentication' archetype into components and classes

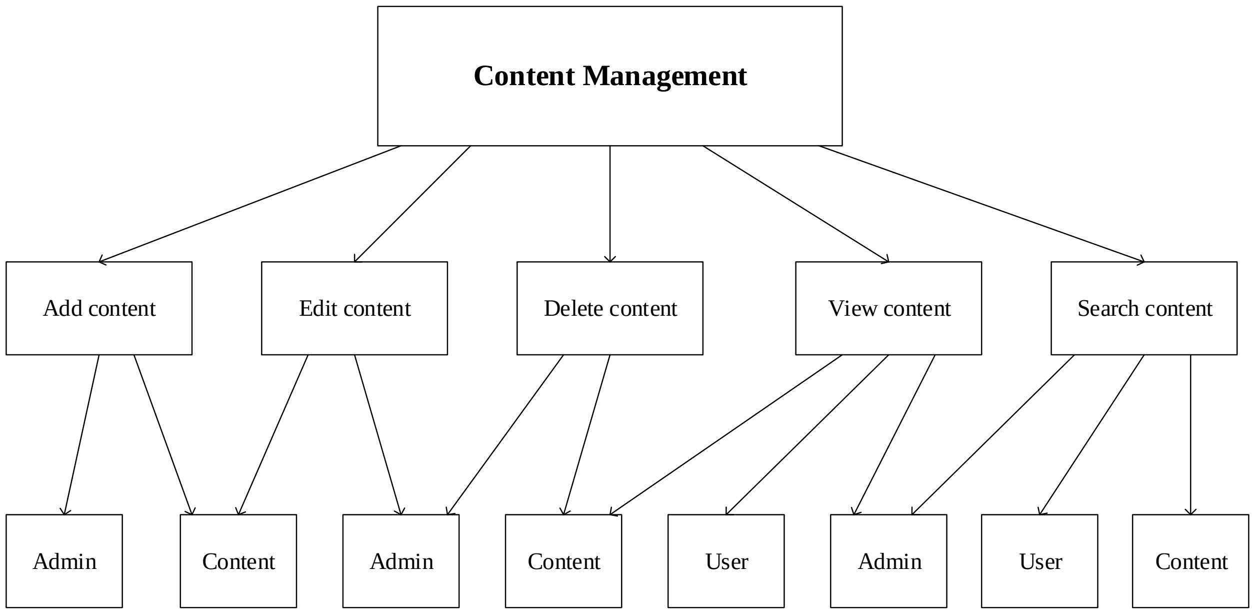


Figure 8. 3: Refining 'Content management' archetype into components and classes

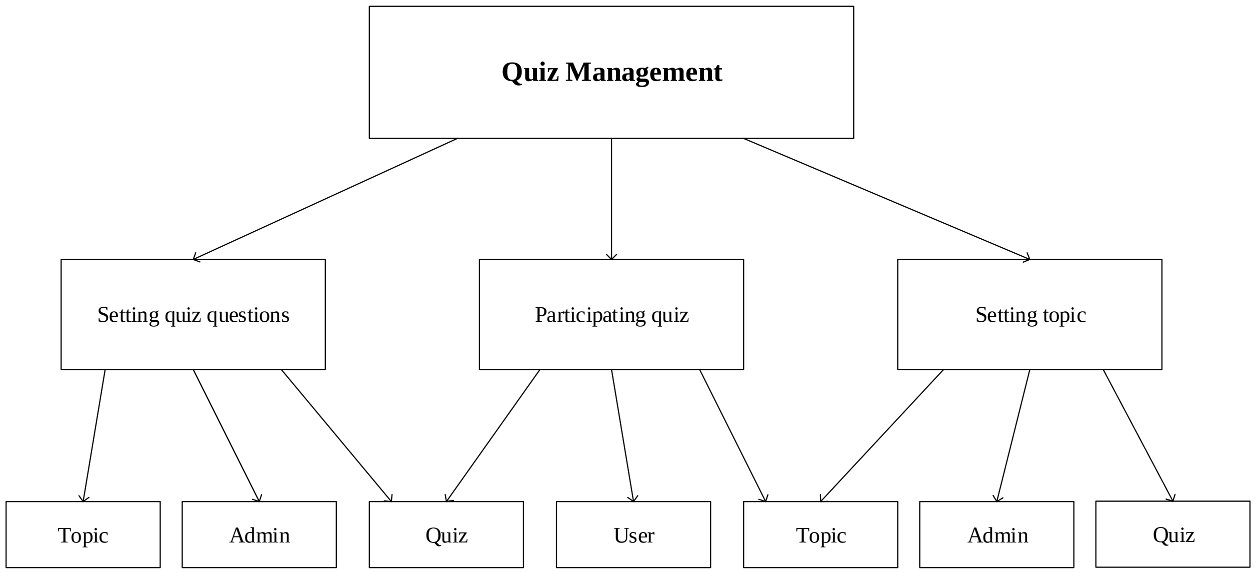


Figure 8. 4: Refining 'Quiz management' archetype into components and classes

Chapter 9

Component Level Design

Step1: Identify all design classes that correspond to the problem domain as defined in the analysis model and architectural model.

The classes are:

1. User
2. Admin
3. Content
4. Topic
5. Quiz

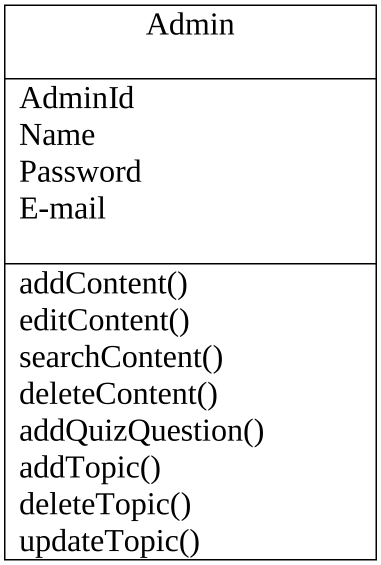
Step2: Identify all design classes that correspond to the infrastructure domain

Such classes that are not belong to the problem domain for our project are:

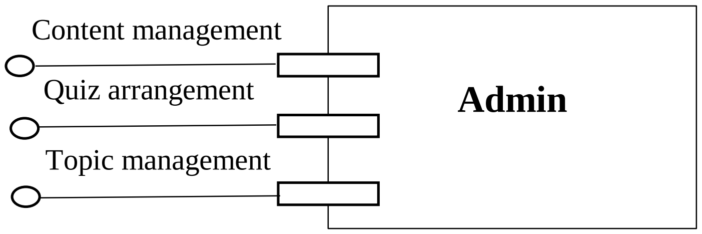
i. DAL

Step3: Elaborate all design classes that are not acquired as reusable components.

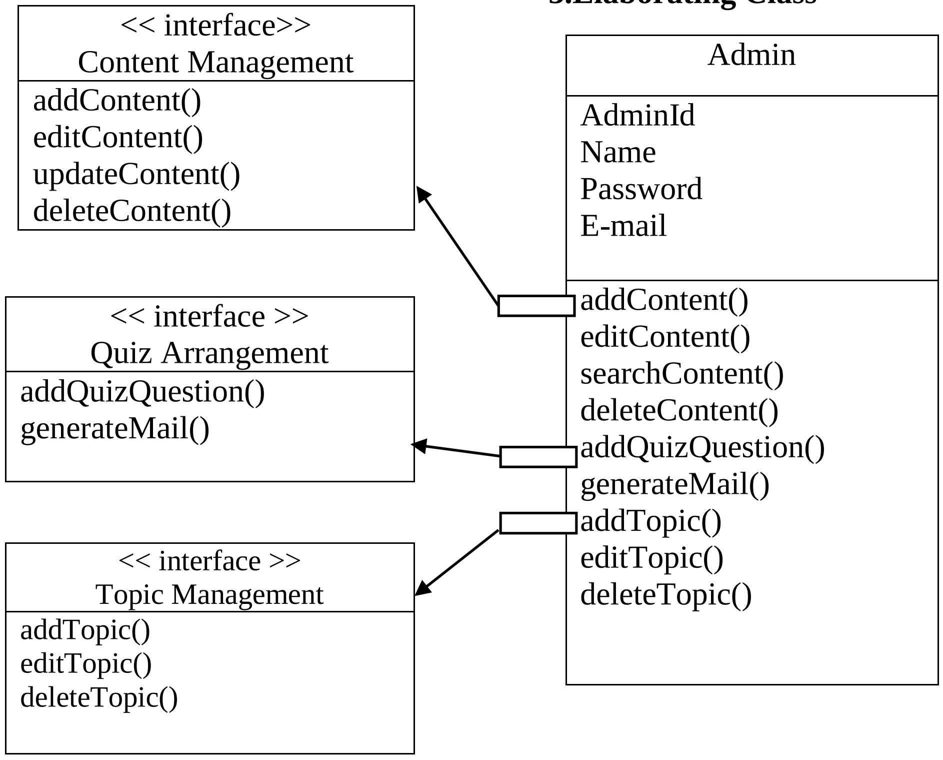
1.Analyzing class



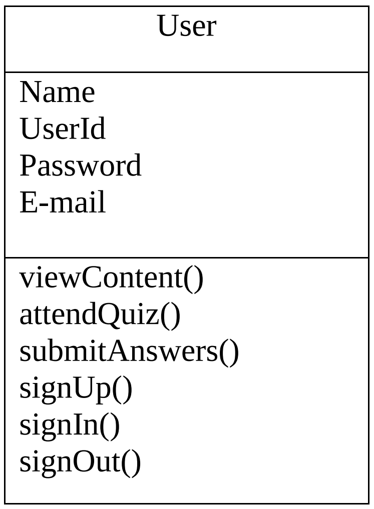
2.Design Component



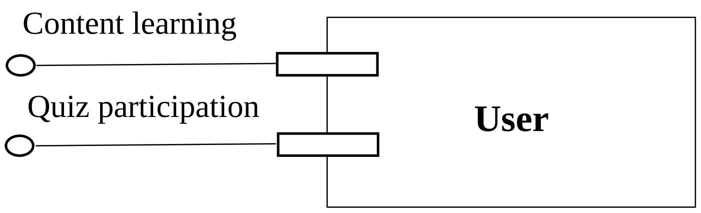
3.Elaborating Class



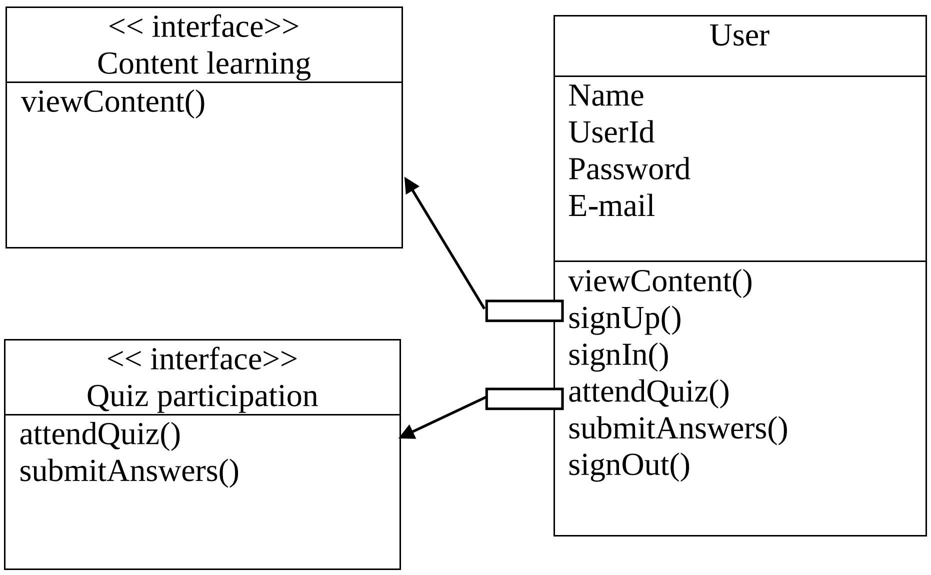
1.Analyzing class



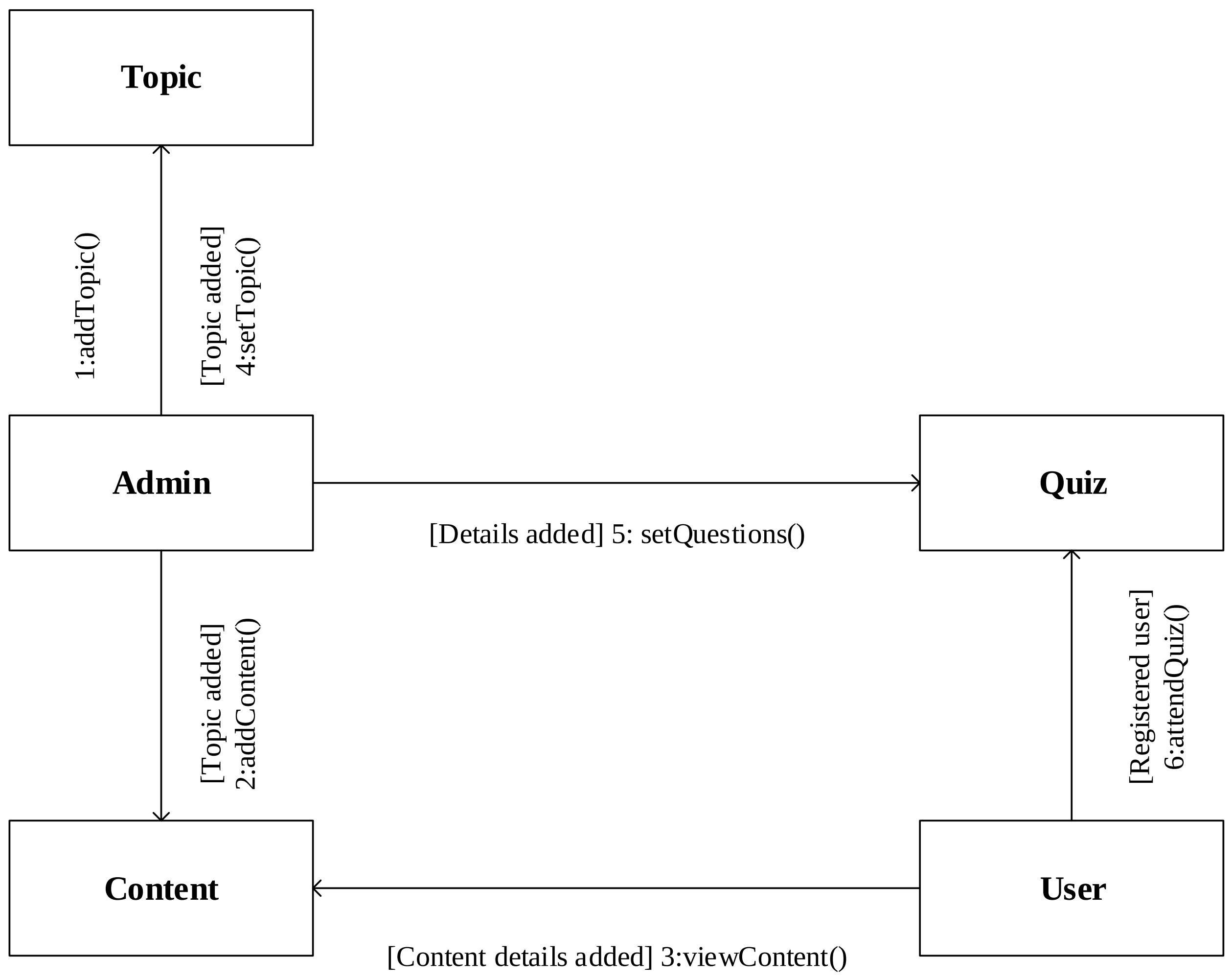
2.Design Component



3.Elaborating Class



Step 3(a): Specify message details when classes or components collaborate



Step 3(b): I think the elaborated classes need not to be refactored any more. As a result, no interface is needed for this design.

Step 3(c): Elaborate attributes and define data types and data structures required to implement them.

User

Name: String=not null {All characters from A-Z}
Email: String = not null {contains value-abc@gmail.com}
Password: String =not null {All characters from A-Z and digits}

Admin

Name: String=null {All characters from A-Z}
Email: String = not null {contains value-abc@gmail.com}
Password: String =not null {All characters from A-Z and digits}

Topic

Name: String=not null {All characters from A-Z}
Description: String=null {All characters from A-Z}

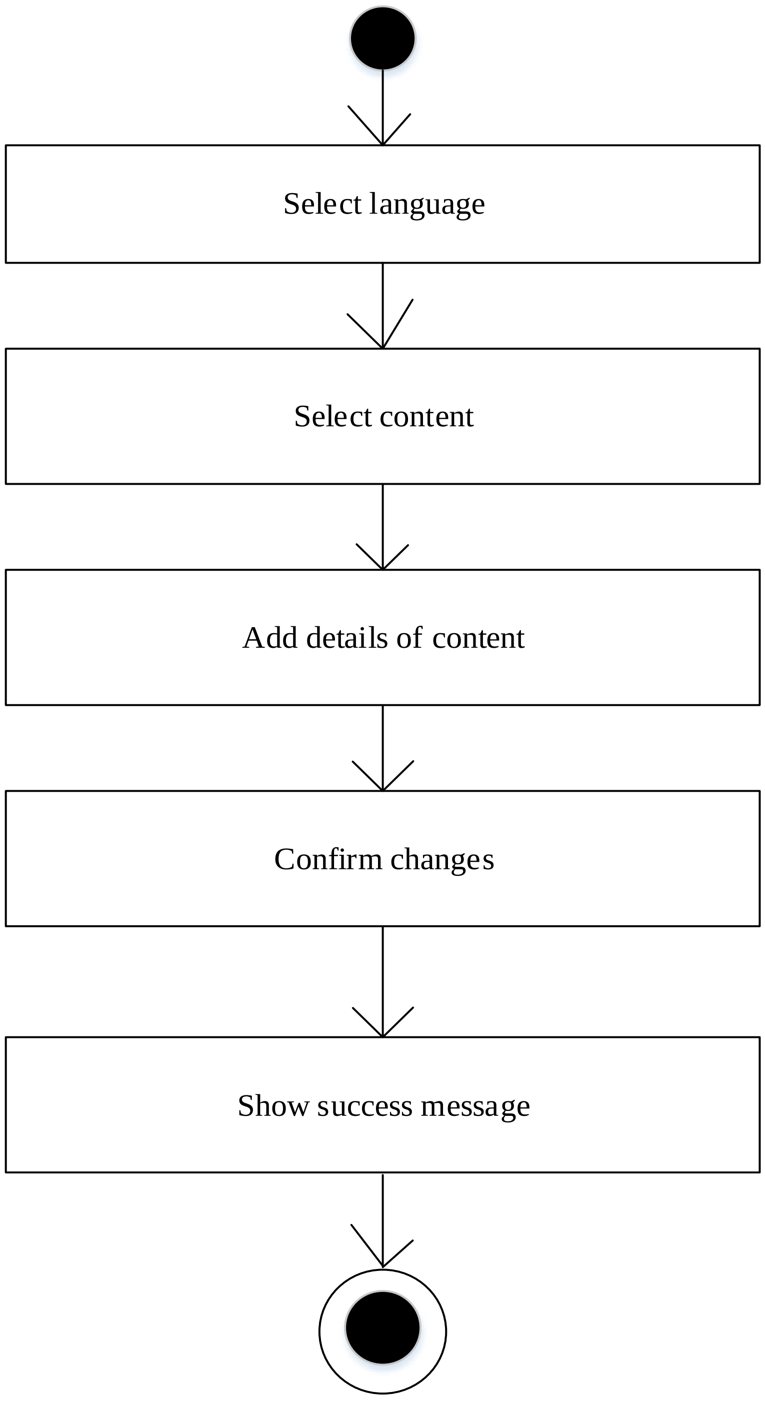
Content

Name: String=not null {All characters from A-Z}
Topic: String=not null {All characters from A-Z}
Details: String=not null {All characters}

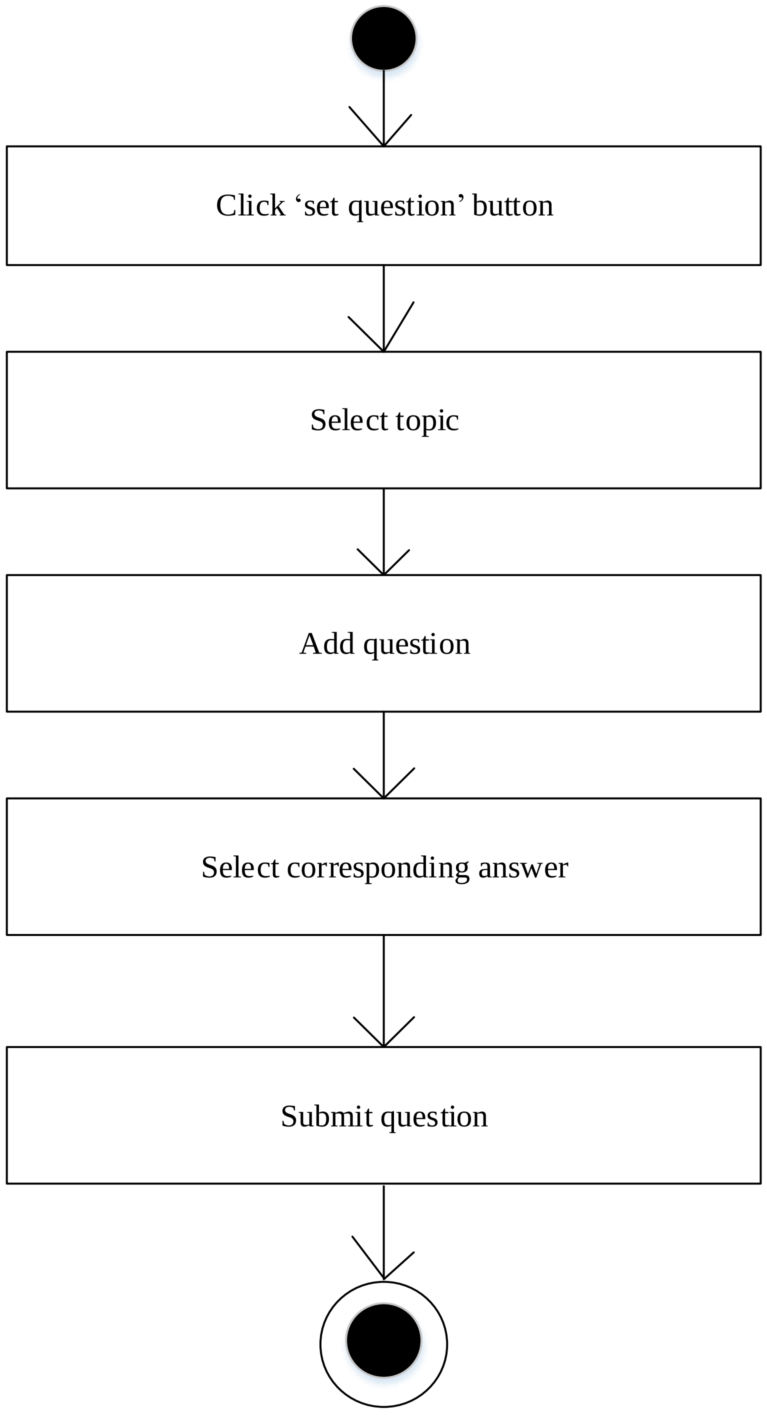
Quiz

Question: String =not null {All characters from A-Z and digits}
Option: String =not null {All characters from A-Z and digits}
Date: String=not null {contains value- “yyyy-MM-dd”}
Answer: not null {“option1”, “option2”, “option3”, “option4”}
Marks: Number = not null {All positive integers}

Step 3(d): Describe processing flow within each operation in detail by means of pseudo code or UML activity diagrams.



Processing flow with UML for operation: Add content details method



Processing flow with UML for operation: Add question method

Step 4:

The persistent data sources (databases and files) and the classes required to manage them are following:

Persistent data source: Database

Classes to manage data source: Database classes for each entity

Step 5:

Develop and elaborate behavioral representations for a class or component.

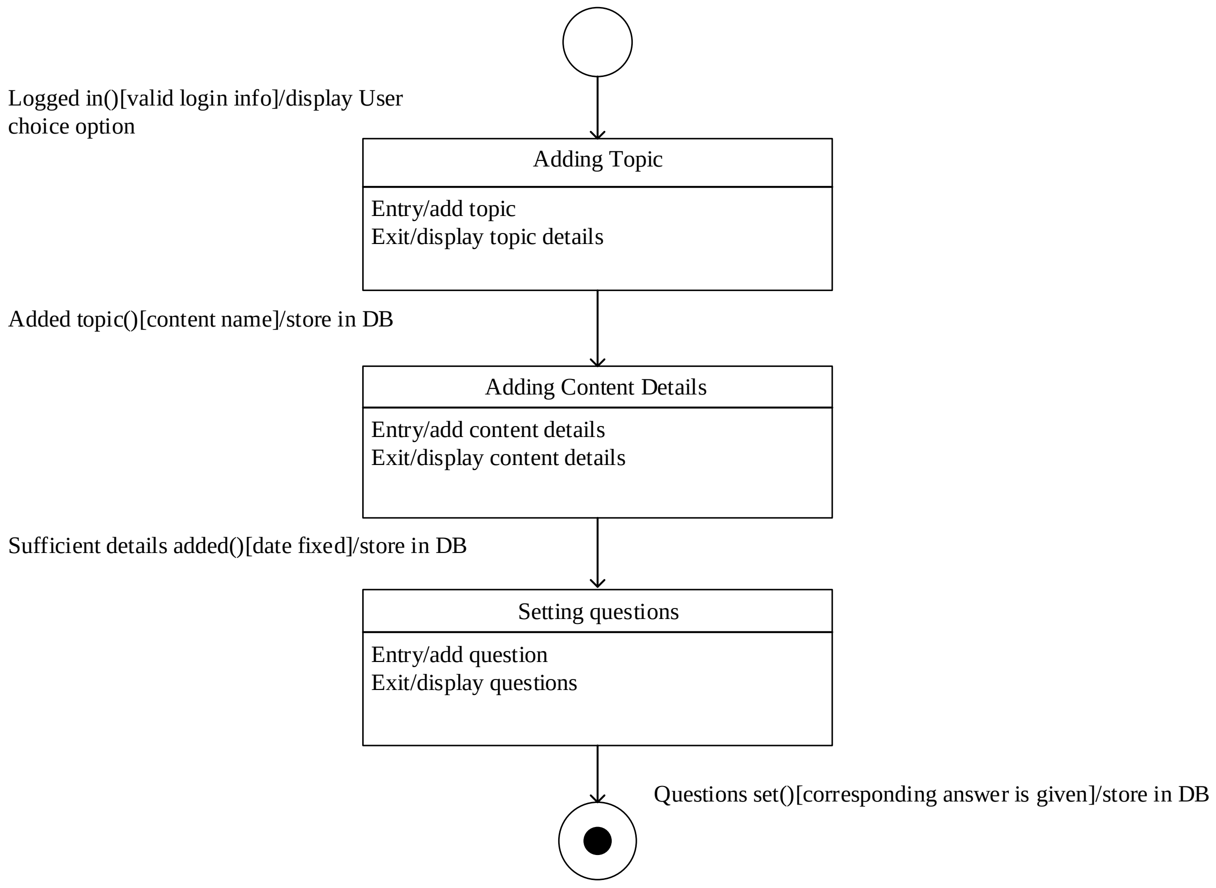

Logged in()[valid login info]/display User choice option
Adding Topic
Entry/add topic
Exit/display topic details
Added topic()[content name]/store in DB
Adding Content Details
Entry/add content details
Exit/display content details
Sufficient details added()[date fixed]/store in DB
Setting questions
Entry/add question
Exit/display questions
Questions set()[corresponding answer is given]/store in DB

Figure 9. 1: State Chart fragment for Admin class

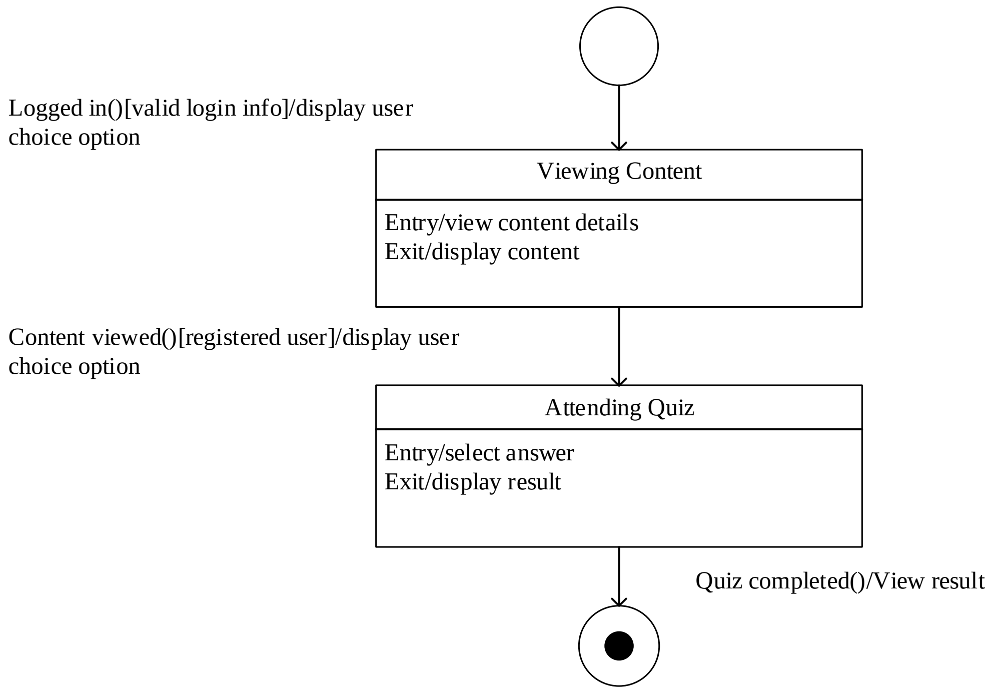
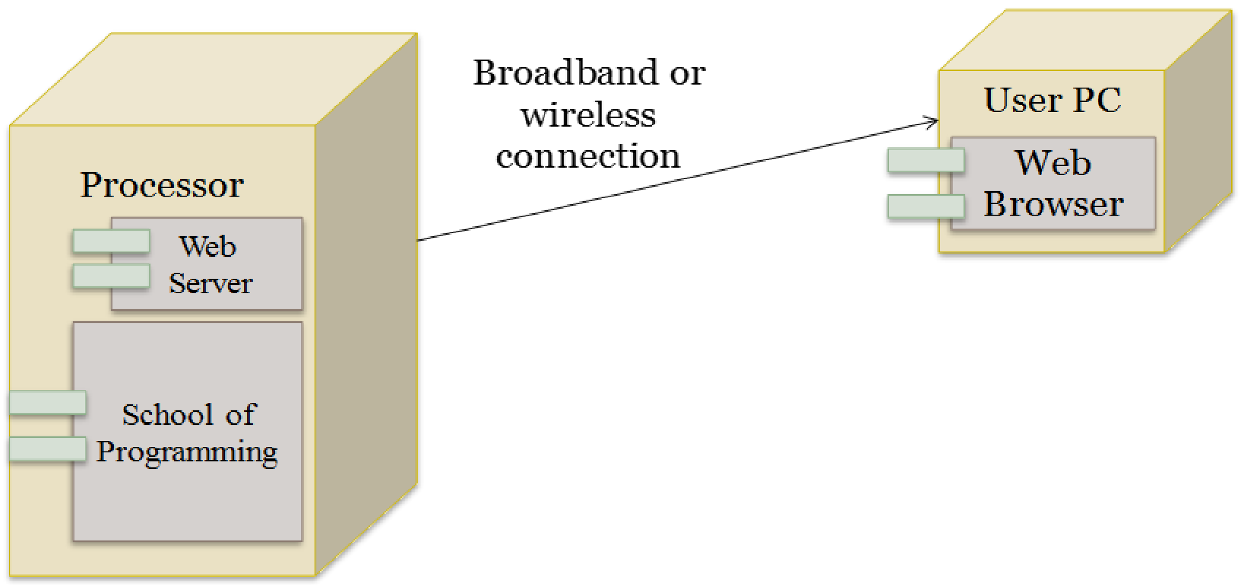

Logged in()[valid login info]/display user choice option
Viewing Content
Entry/view content details
Exit/display content
Content viewed()[registered user]/display user choice option
Attending Quiz
Entry/select answer
Exit/display result
Quiz completed()/View result

Figure 9. 2: State Chart fragment for User class

Step 6: Elaborate deployment diagrams to provide additional implementation detail.



Chapter 10

User Interface Design

10.1 Introduction

User interface design (UI) or is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience. User interface design creates an effective communication medium between a human and a computer. Following a set of interface design principles, design identifies interface objects and actions and then creates a screen layout that forms the basis for a user interface prototype.

10.2 User Analysis

Target User: Users of this system are basically the students the admin.

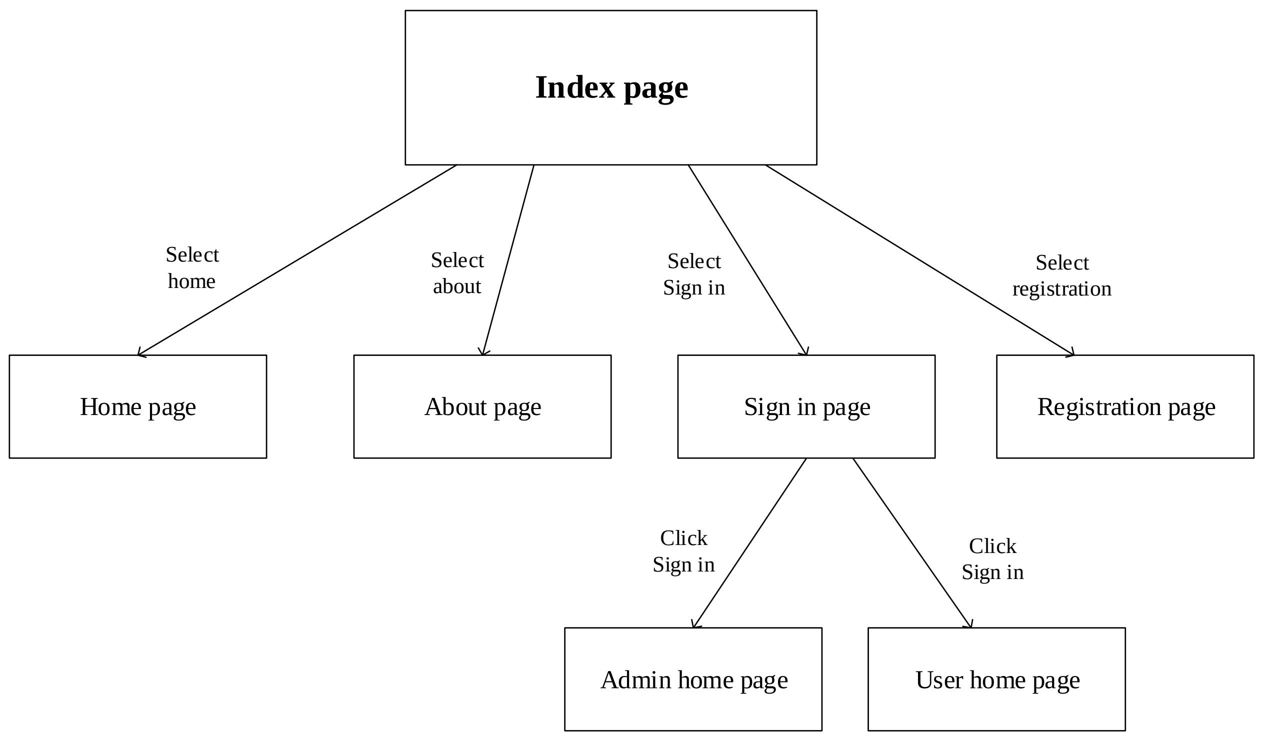
Knowledge Level: Average users should have knowledge of English and internet browsing capability.

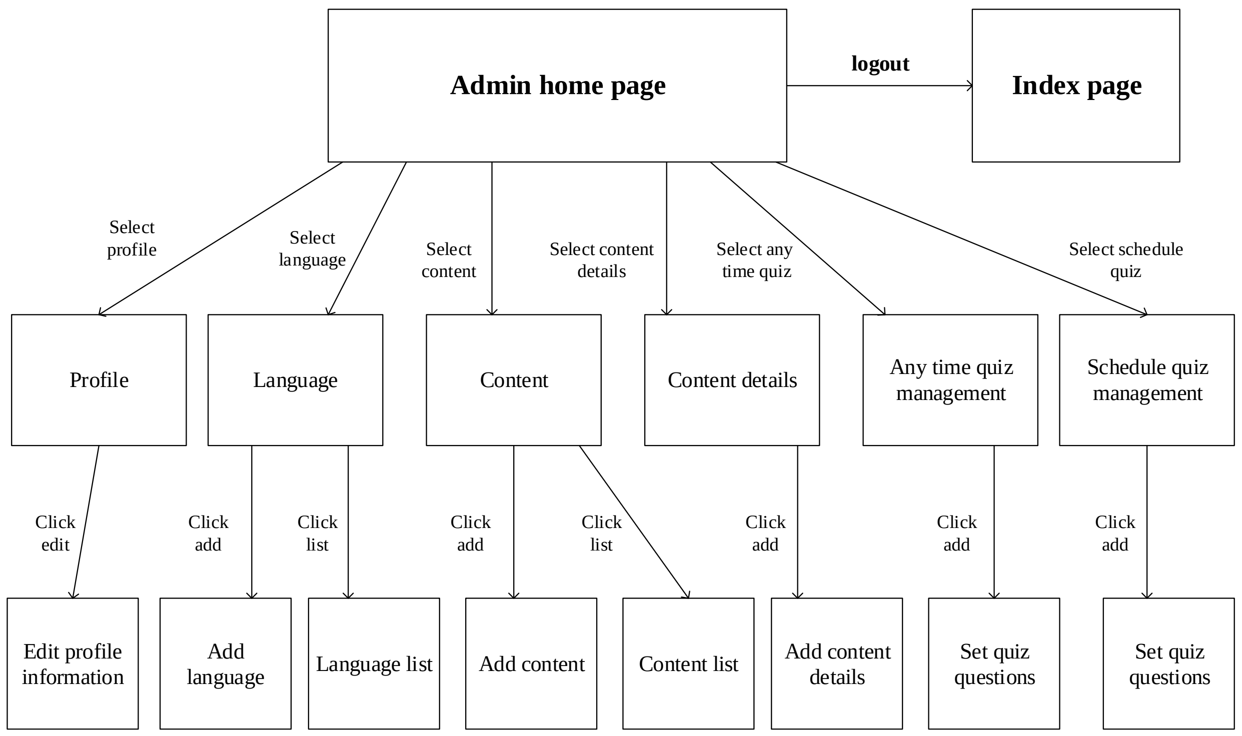
Age range: User Community will be between 15-60.

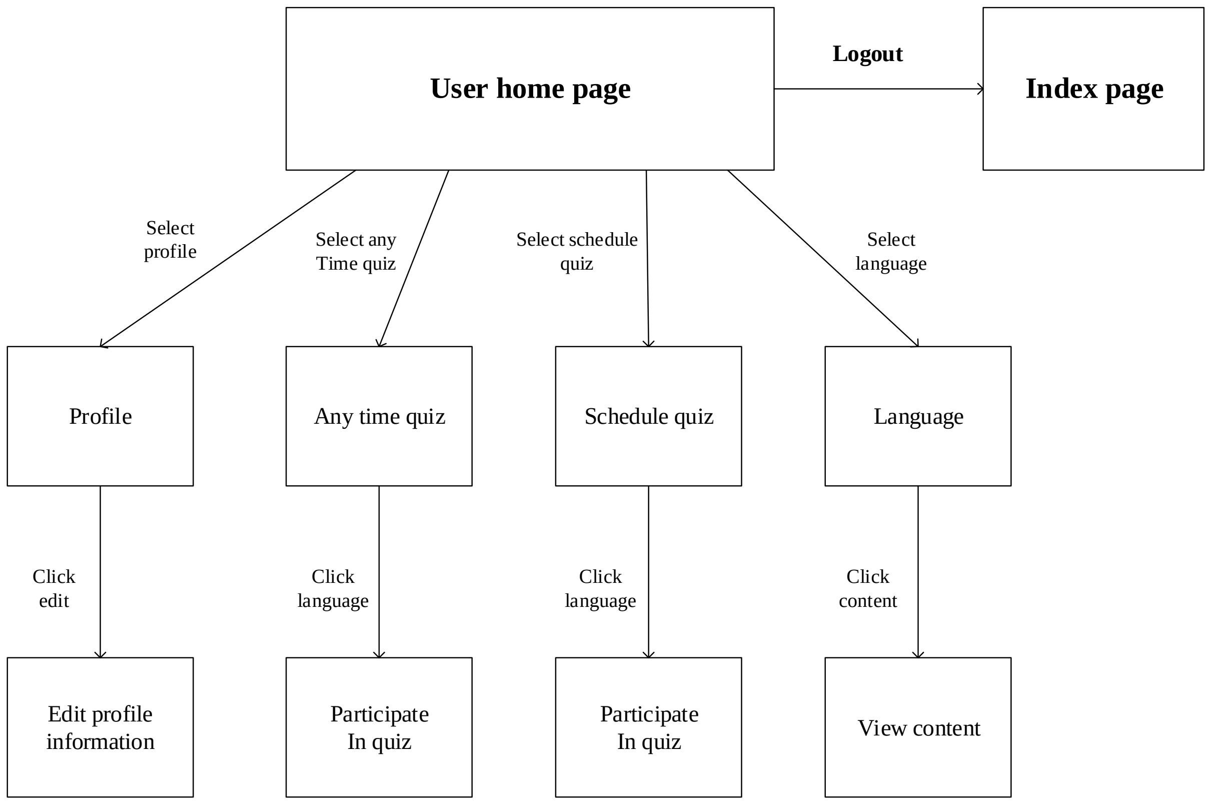
Gender: Both male and female.

Frequency of Use: General Users will use this system frequently.

10.3 Event Transition




Admin home page
logout
Index page
Select profile
Select language
Select content
Select content details
Select any time quiz
Select schedule quiz
Profile
Language
Content
Content details
Any time quiz management
Schedule quiz management
Click edit
Click add
Click list
Click add
Click list
Click add
Click add
Click add
Edit profile information
Add language
Language list
Add content
Content list
Add content details
Set quiz questions
Set quiz questions


User home page
Logout
Index page
Select profile
Select any Time quiz
Select schedule quiz
Select language
Profile
Any time quiz
Schedule quiz
Language
Click edit
Click language
Click language
Click content
Edit profile information
Participate In quiz
Participate In quiz
View content

This is the initial interface of ‘School of Programming’. A user will see this interface whenever he/she will enter the system.

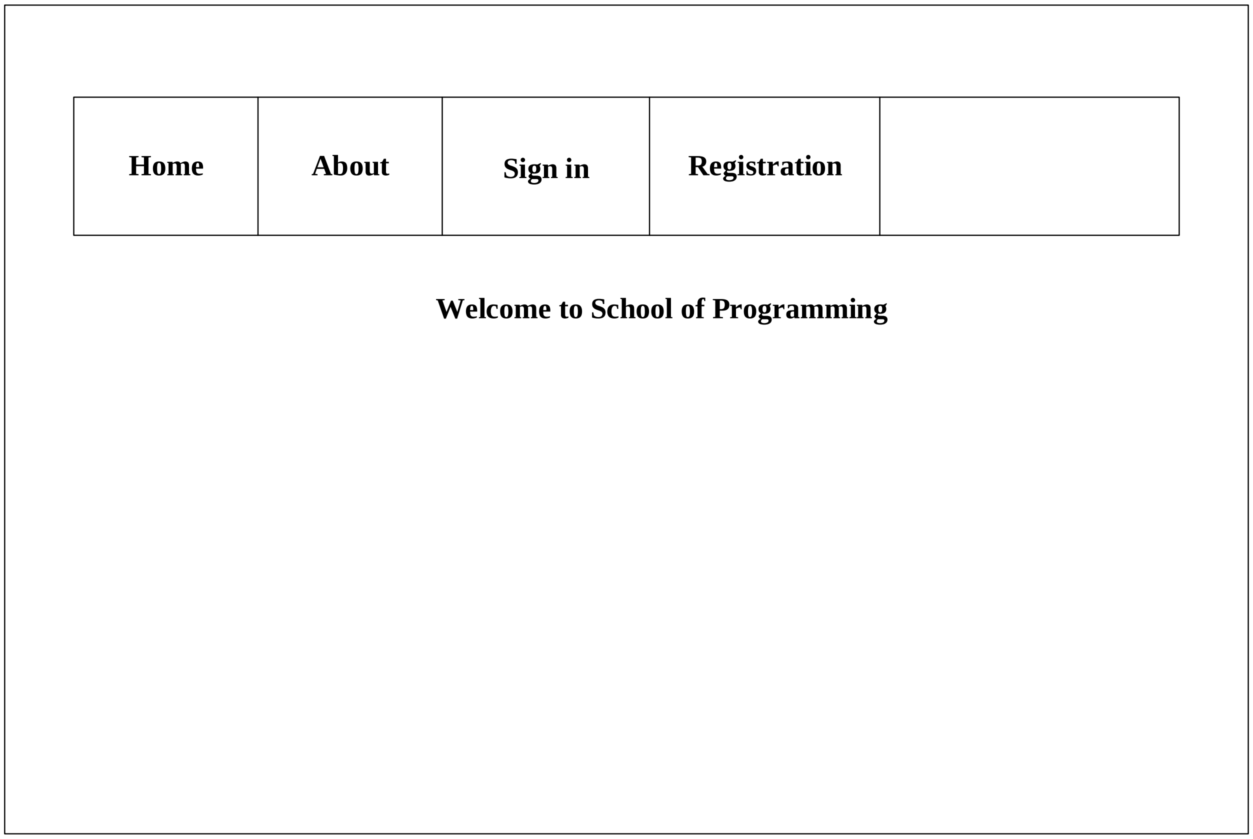


Figure 10. 1: User Interface of Home page

This is the **‘registration’** interface of ‘School of Programming’. A user can create new account through registration. The user needs to fill up the given field and submit.

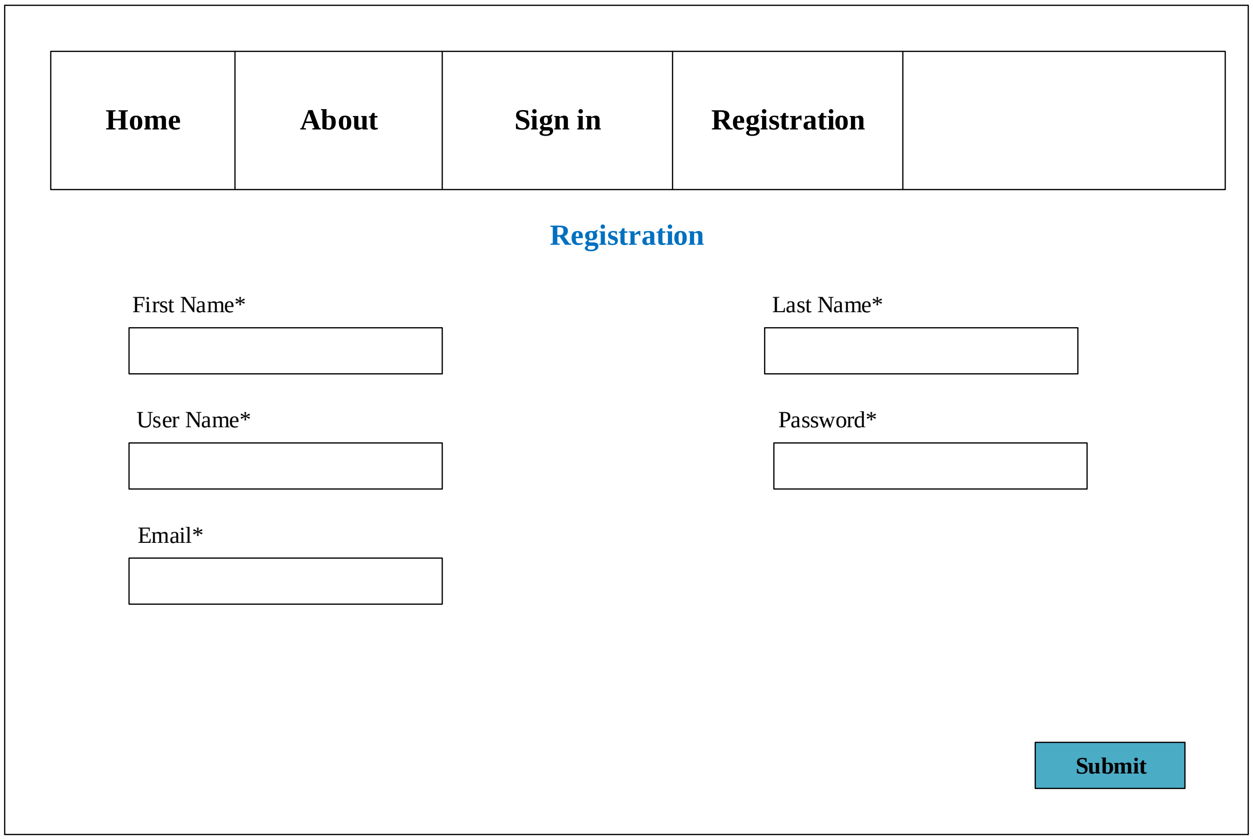


Figure 10. 2: User Interface of ‘Registration’ page

This is the **‘sign in’** interface of ‘School of Programming’ where a registered user can sign in by his email and password.

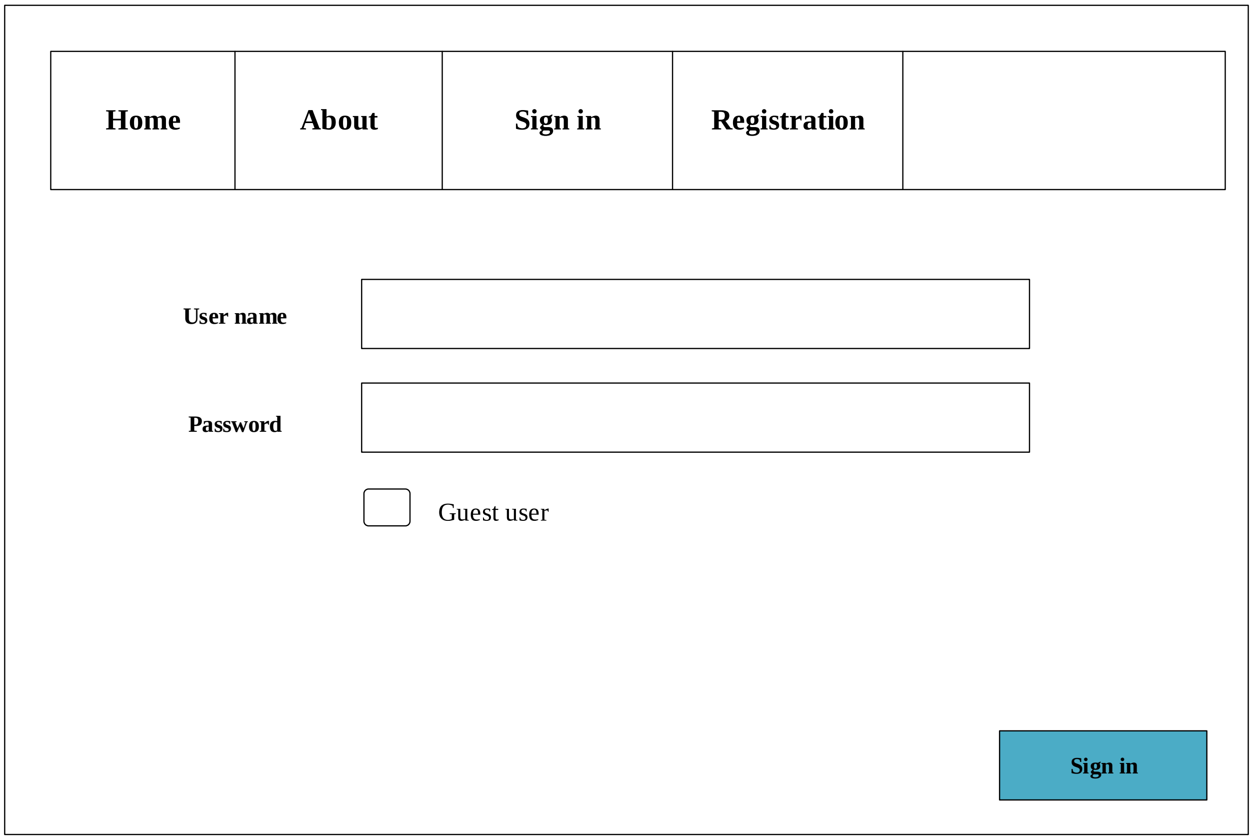


Figure 10. 3: User Interface of ‘Sign in’ page

After login, the home page of admin will be displayed which will look like the following figure:

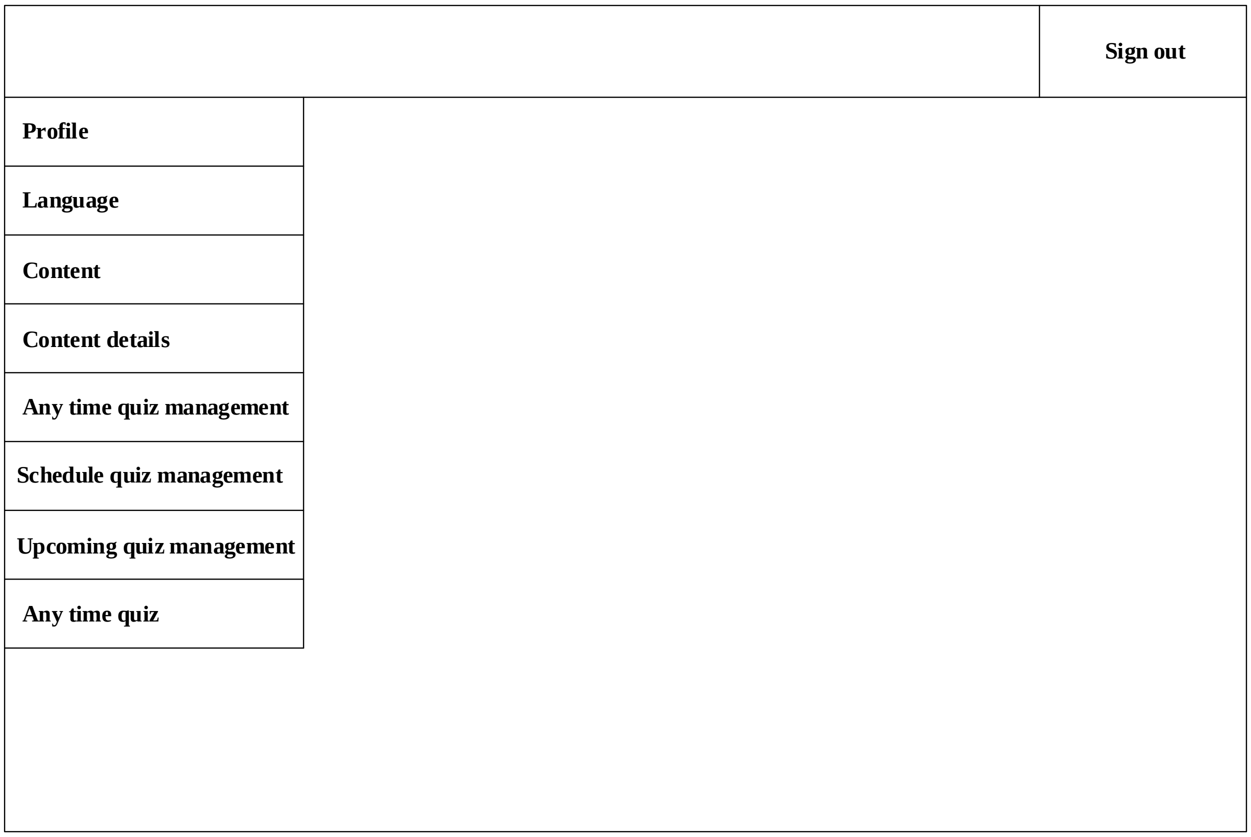


Figure 10. 4: User Interface of Admin home page

This is the **‘add language’** interface of ‘School of Programming’. The following figure illustrates how admin will add language in the system:

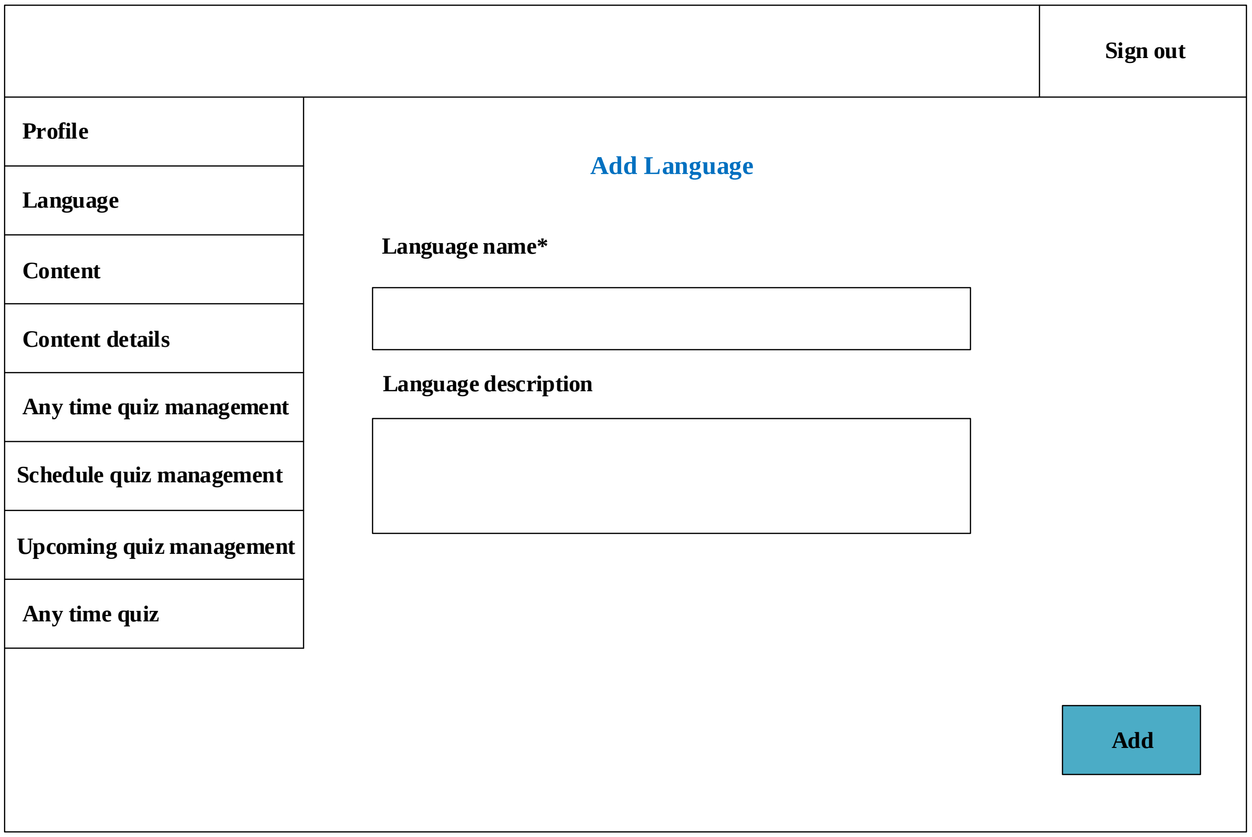


Figure 10. 5: User Interface of ‘Add language’ page

This is the **'add content'** interface of 'School of Programming'. The following figure illustrates how admin will add content in the system:

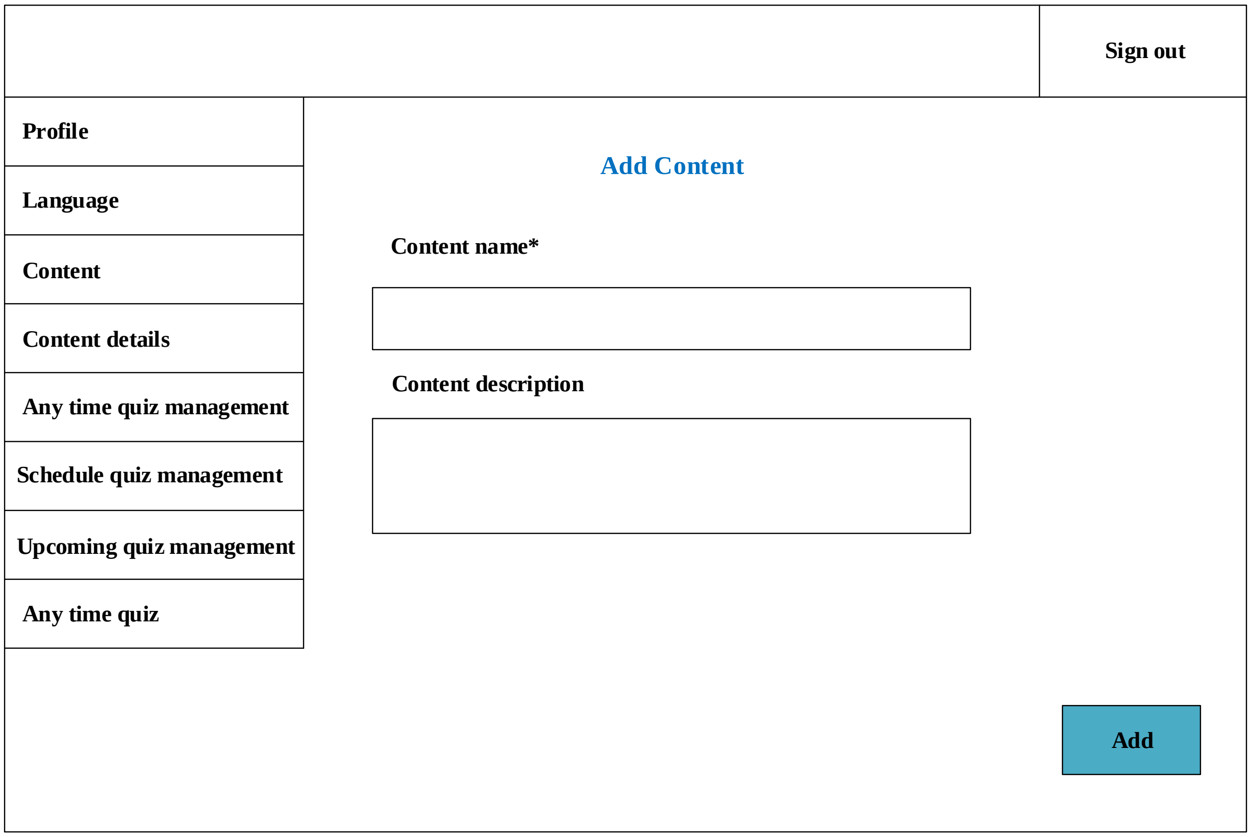


Figure 10. 6: User Interface of 'Add content' page

This is the **‘add content details’** interface of ‘School of Programming’. The following figure illustrates how admin will add content details in the system:

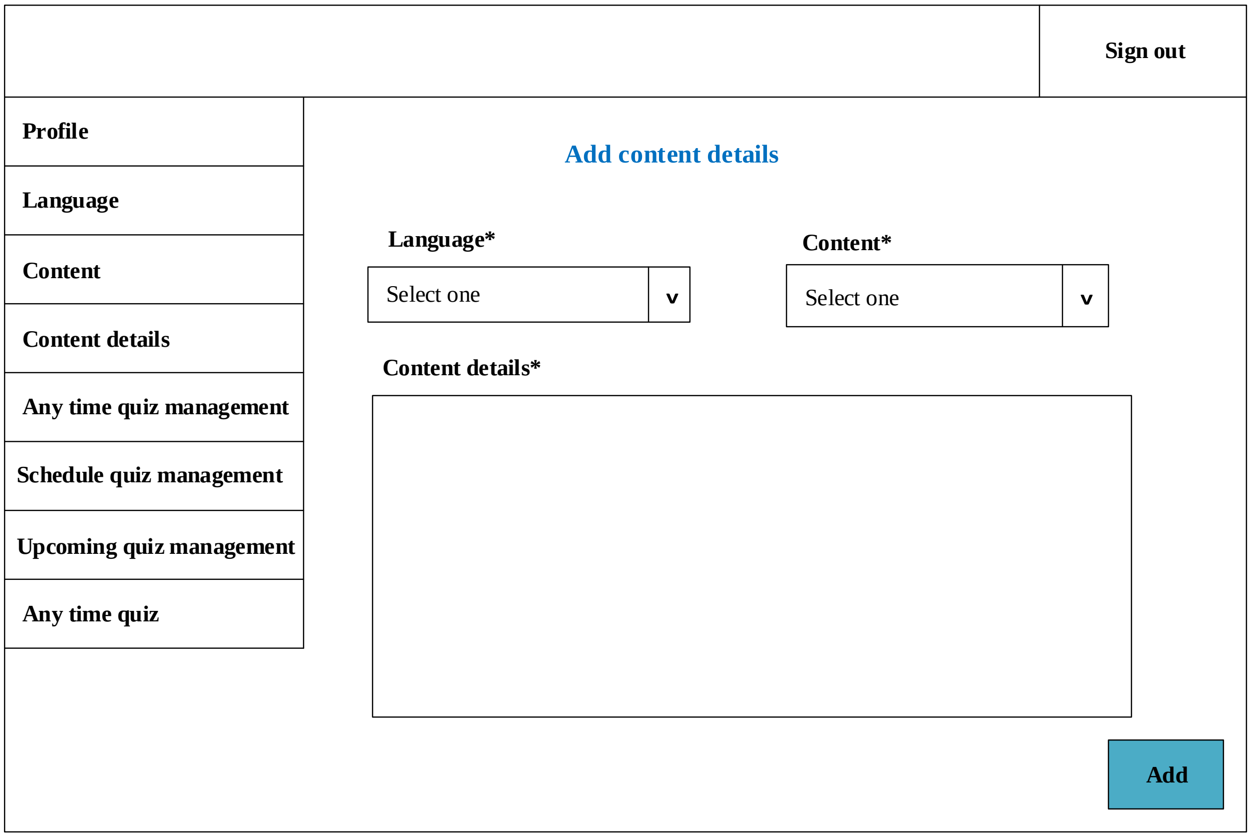


Figure 10. 7: User Interface of ‘Add content details’ page

This is the **‘set quiz question** interface of ‘School of Programming’. The following figure illustrates how admin will add quiz questions in the system:

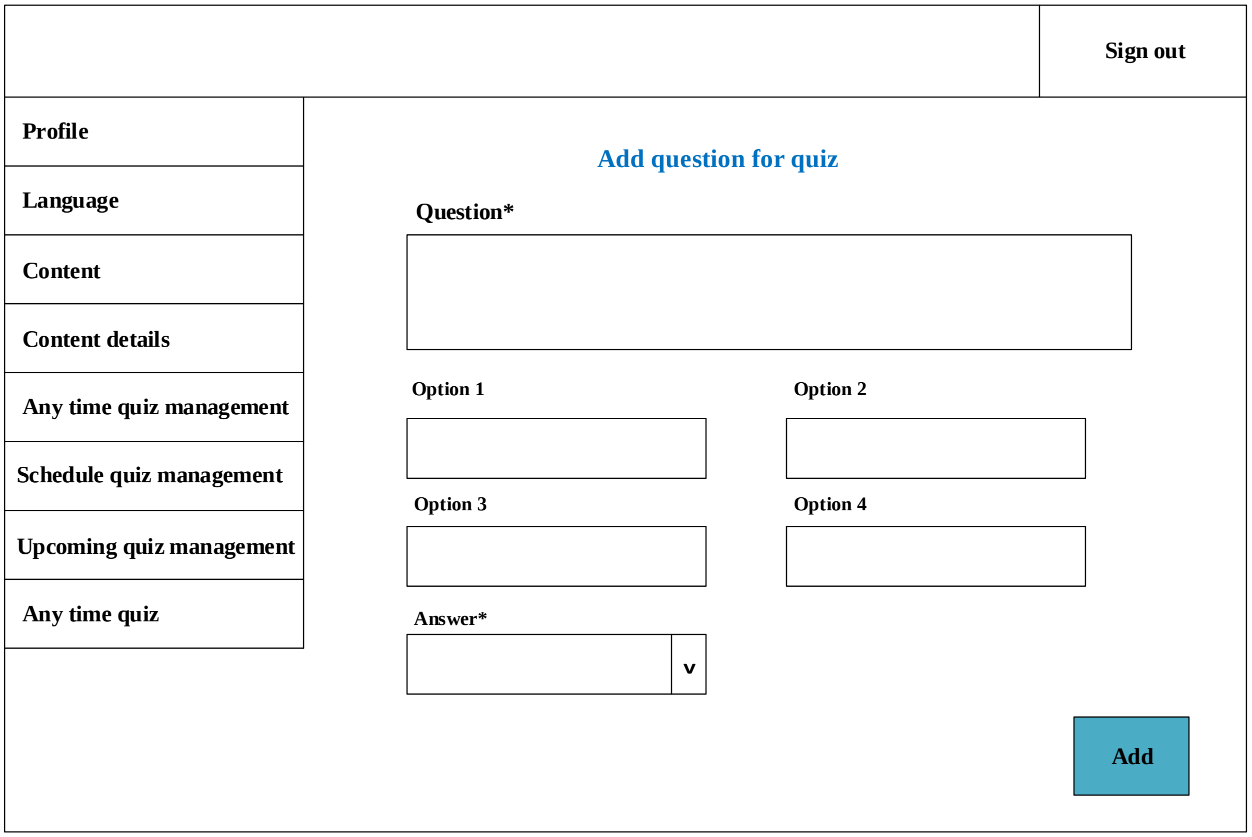


Figure 10. 8: User Interface of ‘set quiz question’ page

This is the **‘arranging schedule quiz’** interface of ‘School of Programming’. The following figure illustrates how admin will arrange schedule quiz in the system:

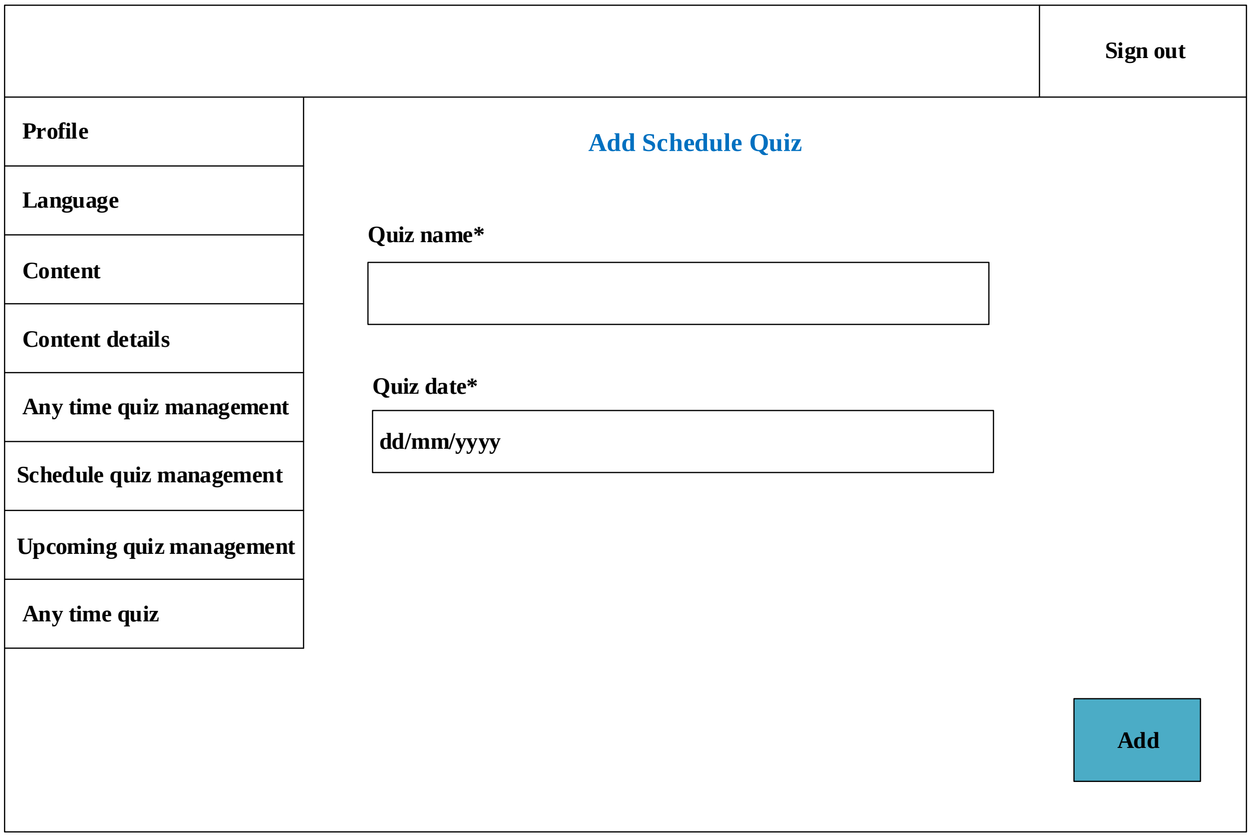


Figure 10. 9: User Interface of ‘arrange schedule quiz’ page

After login, the home page of user will be displayed which will look like the following figure:

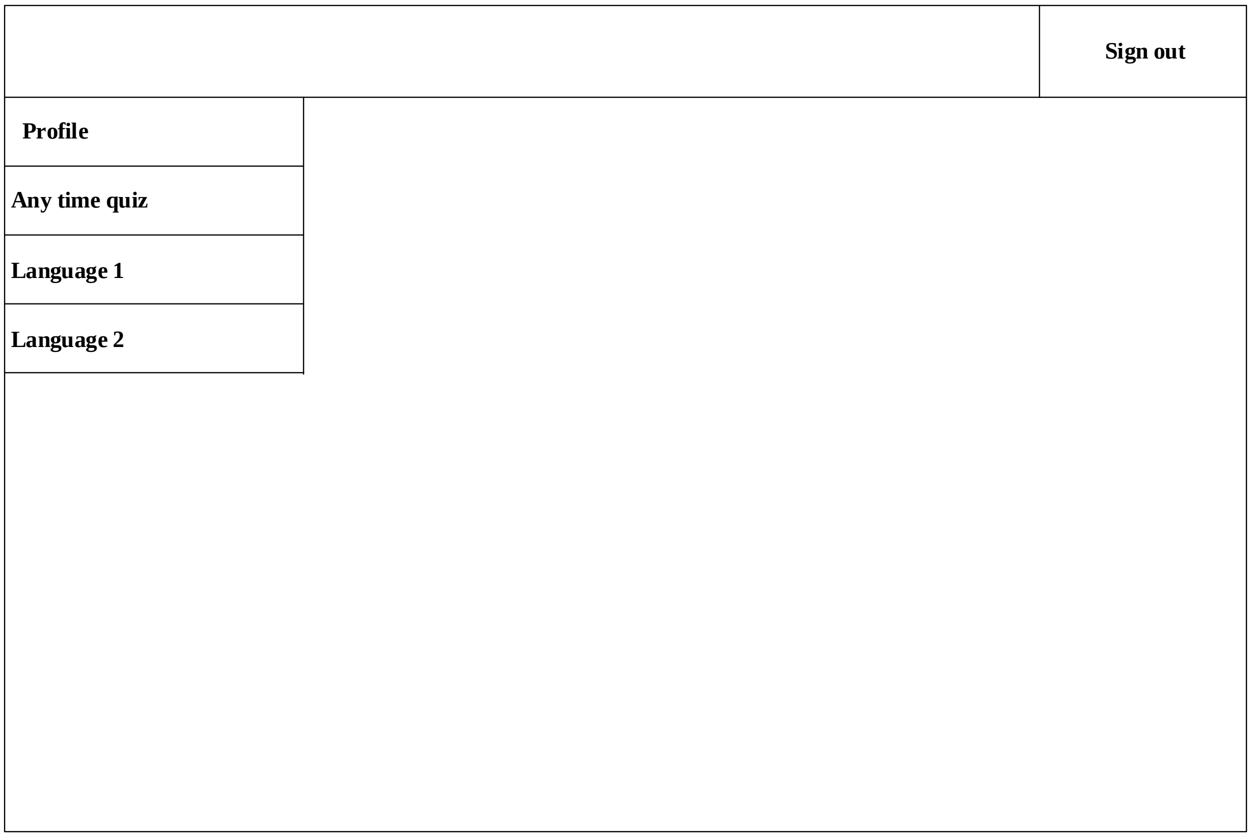


Figure 10. 10: User Interface of User home page

This is the **'quiz'** interface of 'School of Programming'. The following figure illustrates how quiz interface will appear to the general user in the system:

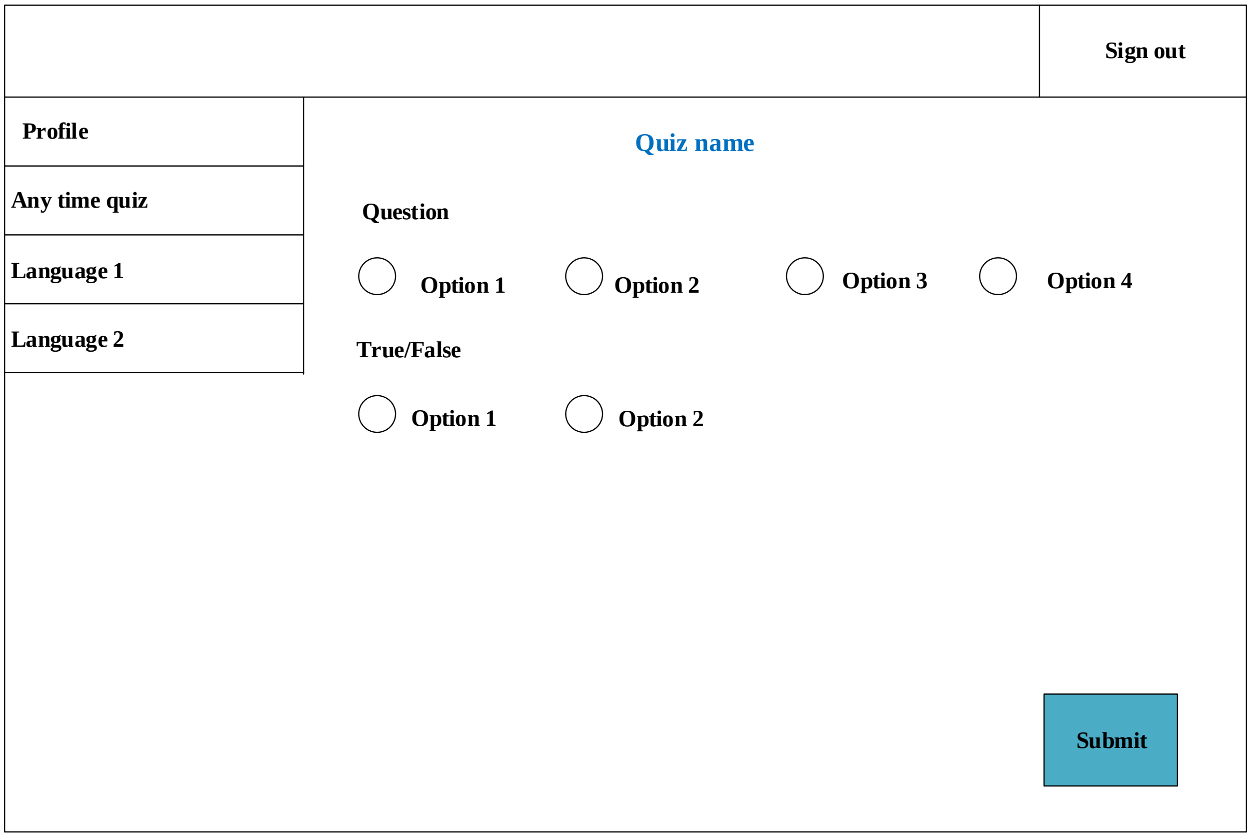


Figure 10. 11: User Interface of 'Quiz' page

The home page of guest user will look like the following figure:

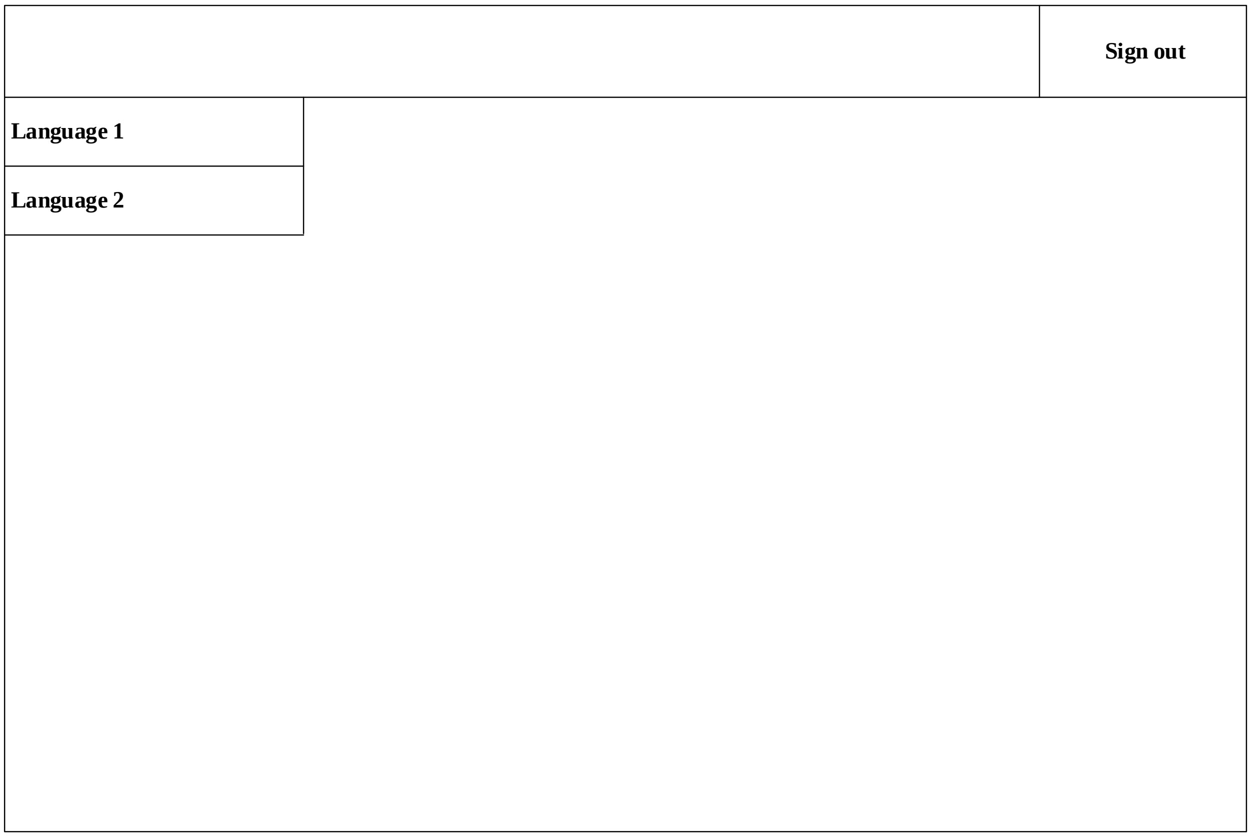


Figure 10. 12: User Interface of Guest user home page

This is the **‘manage profile’** interface of ‘School of Programming’. The following figure illustrates how general user and admin will manage profile in the system:

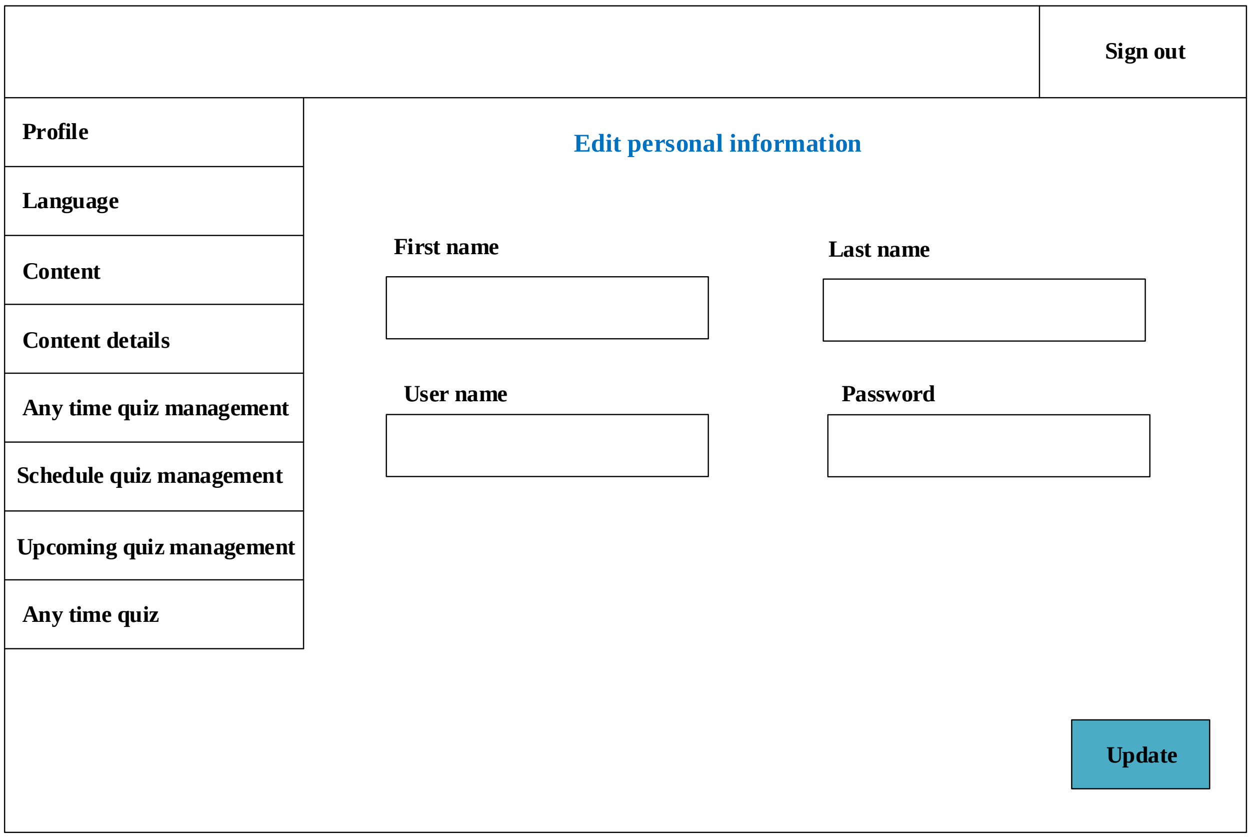


Figure 10. 13: User Interface of ‘Manage profile’ page

Chapter 11

Testing

Version	Date	Document name
1.0	20-12-2017	Test plan outline

11.1 Introduction

This test plan describes the testing approach and overall framework that will drive the testing of the School of Programming version 1.0. The Test Plan document documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project's product. The Test Plan document is created during the Planning Phase of the project. Its intended audience is the project supervisor, developer and testing team. Some portions of this document may on occasion be shared with the client/user and other stakeholder whose input/approval into the testing process is needed.

11.2 Test plan identifier

This is the Master Test Plan of the project. As the software will be released once, I will stick to the master test plan. Besides how much testing will be performed is dependent on the work progress and the deadline.

11.2.1 Summary of items and features to be tested

Web application version is to be tested for School of Programming. In this application, it has some features that to be tested, these are: user authentication, content management, quiz management, setting questions and submitting answers.

11.2.2 Requirement and history of items

The requirement of items that will be tested that are collected from the software requirement and specification (SRS) and design primarily.

11.2.3 High-level description of testing goals

Testing makes a software bug free and more secure. After releasing a software product, there may have a lot of bugs which may not be solved during the development phase. These bugs are solved after testing phase. So after development, testing is required for developing a better software product.

11.2.4 Reference Document

- Software Requirement Specification
- Architectural design
- Component design
- User Interface design

11.3 Test items

- Web application of School of Programming project.

11.4 Features to be tested

- Registration
- Sign in
- Add new language
- Add new content
- Add content details
- Edit language
- Edit content
- Edit content details
- Delete language
- Delete content
- Search language
- Search content
- Add questions
- Submit answers
- Add schedule quiz
- Sign out

11.5 Features not to be tested

- All the features of this system will be tested

11.6 Approach

- I will cover only black box testing for my project.
- I will test the functionality of the application.
- I will test all the features in Google Chrome and Mozilla Firefox browser only.
- Integration testing will be performed.

11.7 Item pass/fail criteria

Specifying the criteria that I will use to determine whether each test item of my project has passed or failed during testing. The planning criteria gives the framework for how the system will be evaluated and under what circumstances it will be released.

11.8 Suspension criteria and Resumption requirements

Suspension criteria will be used when it is needed to suspend all or a portion of the testing activities when the testing has no value and the build is not working properly which is overall a wasting of resources. On the other hand, resumption criteria specify when testing can be resumed after it has been suspended. These will be applied in such situations:

Suspension when-

- A defect is introduced that doesn't allow any further testing.
- Unavailability of external dependent systems during execution.

Resumption when-

- When the defect is fixed.
- When the external dependent systems become available again.

11.9 Test Deliverables

I will provide the following deliverables after testing at the end of the project:

- Test plan document
- Test cases

11.10 Environmental needs

- As we need data for testing, the test data which will be provided is an environmental need. Without the provided data, we cannot perform any phase of testing.
- I will conduct an integration test after testing the individual features.

11.11 Staffing and training needs

I will take some short training on how to perform functional and integration testing as I am not very proficient in functional and integration testing.

11.12 Responsibilities:

Responsibility for testing different modules of the testing is given below:

Serial number	Module	Responsible	Roll
1.	User authentication	Md. Abu Bakar Siddique	BSSE0609
2.	Content management	Md. Abu Bakar Siddique	BSSE0609
3.	Any time quiz management	Md. Abu Bakar Siddique	BSSE0609
4.	Schedule quiz management	Md. Abu Bakar Siddique	BSSE0609
5.	Setting question	Md. Abu Bakar Siddique	BSSE0609

11.13 Risks and contingencies:

- If development takes much time to finish than estimated, then testing will be late. Consequently, there is a risk of not meeting the deadline of the software.
- In every software project, there is a chance of changing or modification of the requirements. If this happens, I may need to redesign or modify my plan and test cases.

If the requirement will change further, the following steps will be taken:

- Since I have to complete my product in due time, I will reschedule my working period and increase my working time to complete testing.
- The number of test cases may be reduced.
- Number of acceptable defects may be increased. These defects will be fixed in further releases.

11.14 Testing cost:

Since this is an academic project, so there is no need to estimate or consider the cost of testing.

11.15 Approval

Name	Role	Approver/Reviewer	Approval date
Dr. Md. Shariful Islam	Project Supervisor		

11.16 Test Cases

Test Case-1:

Test case Name: Registration

Short Description: Test registration system

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test with an empty required field with other fields filled up validly and click submit	User is not registered	The system will display “Please fill out this field”	pass	
2.	Test with filling one field validly and other fields empty and click submit	User is not registered	The system will display “Please fill out this field”	pass	
3.	Test with filling some fields and other fields empty and click submit	User is not registered	The system will display “Please fill out this field”	pass	
4.	Test with first name, last name, username, email, password where email is invalid and click submit	User is not registered	The system will display an error message “invalid email address”	pass	
5.	Test with existing username and click submit	User is not registered	The system will display an error message “Name already exists, please try another one”	pass	
6.	Test with all valid information and click submit	User is not registered	Successfully registered in the system and display an error message “You are successfully registered”	pass	

Test Case-2:

Test case Name: Sign in

Short Description: Test sign in system

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test with empty user name, empty password and click submit button	Users are not signed in	The system will display "Invalid user name & password"	pass	
2.	Test with valid user name and empty password and click submit	Users are not signed in	The system will display "Invalid user name & password"	pass	
3.	Test with valid user name and wrong password and click submit	Users are not signed in	The system will display "Invalid user name & password"	pass	
4.	Test with empty user name and valid password and click submit	Users are not signed in	The system will display "Invalid user name & password"	pass	
5.	Test with wrong user name and valid password and click submit	Users are not signed in	The system will display "Invalid user name & password"	pass	
6.	Test with wrong user name and wrong password and click submit	Users are not signed in	The system will display "Invalid user name & password"	pass	
7.	Test if password field is masked	Users are not signed in	Displays characters in bullets	pass	
8.	Give correct email and correct password and click submit	Users are not signed in	Redirect to authenticated users home page	pass	

Test Case-3:

Test case Name: Add language

Short Description: Test of adding new language

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test without adding any language name and click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
2.	Test with adding language name and description, then click add button	Must be signed in as admin	The system will display “successfully added language information”	pass	
3.	Test with adding language name and empty description, then click add button	Must be signed in as admin	The system will display “successfully added language information”	pass	

Test Case-4:

Test case Name: Add Content

Short Description: Test of adding new content

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test without adding any content name and click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
2.	Test with adding content name and description, then click add button	Must be signed in as admin	The system will display “successfully added content information”	pass	
3.	Test with adding content name and empty description, then click add button	Must be signed in as admin	The system will display “successfully added content information”	pass	

Test Case-5:

Test case Name: Add Content details

Short Description: Test of adding content details

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test with empty language, empty content and content details, then click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
2.	Test with empty language, filled content and content details, then click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
3.	Test with empty content, filled language and content details, then click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
4.	Test with empty content details, filled language and content, then click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
5.	Test with empty language and empty content, and filled content details, then click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
6.	Test with filled language, content and content details, then click add button	Must be signed in as admin	The system will display “successfully added content details”	pass	

Test Case-6:

Test case Name: Edit language

Short Description: Test of editing language

Test steps	**Action**	**Pre-condition**	**Expected system response**	**Pass/Fail**	**Comments**
1.	Test with clicking ‘back’ button	Must be signed in as admin	Redirect to language list page	pass	
2.	Test without adding any language name and click update button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
3.	Test without editing language name and click update button	Must be signed in as admin	The system will display “successfully updated language information”	pass	
4.	Test after editing language name and click update button	Must be signed in as admin	The system will display “successfully updated language information”	pass	

Test Case-7:

Test case Name: Edit Content

Short Description: Test of editing content

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test with clicking ‘back’ button	Must be signed in as admin	Redirect to content list page	pass	
2.	Test without adding any content name and click update button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
3.	Test without editing content name and click update button	Must be signed in as admin	The system will display “successfully updated content information”	pass	
4.	Test after editing content name and click update button	Must be signed in as admin	The system will display “successfully updated content information”	pass	

Test Case-8:

Test case Name: Edit Content details

Short Description: Test of adding content details

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test without editing language, content and content details, then click update button	Must be signed in as admin	The system will display “successfully updated content details”	pass	
2.	Test with empty language, empty content and content details, then click update button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
3.	Test with empty language, filled content and content details, then click update button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
4.	Test with empty content, filled language and content details, then click update button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
5.	Test with empty content details, filled language and content, then click update button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
6.	Test with filled language, content and edit content details, then click update button	Must be signed in as admin	The system will display “successfully added content details”	pass	

Test Case-9:

Test case Name: Delete language

Short Description: Test of deleting language

Test steps	**Action**	**Pre-condition**	**Expected system response**	**Pass/Fail**	**Comments**
1.	Select a language to delete, then click delete button	Must be signed in as admin	The system will display "successfully deleted language information"	pass	

Test Case-10:

Test case Name: Delete content details

Short Description: Test of deleting content details

Test steps	**Action**	**Pre-condition**	**Expected system response**	**Pass/Fail**	**Comments**
1.	Select a content details to delete, then click delete button	Must be signed in as admin	The system will display "successfully deleted content details"	pass	

Test Case-11:

Test case Name: Search language

Short Description: Test of searching language

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test with empty search box	Must be signed in as admin	The system will display all the languages	pass	
2.	Search an existing language in search box	Must be signed in as admin	The system will display the existing language	pass	
3.	Search a language that does not exist	Must be signed in as admin	No matching records found	pass	
4.	Search with wrong language name	Must be signed in as admin	No matching records found	pass	

Test Case-12:

Test case Name: Search content

Short Description: Test of searching content

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test with empty search box	Must be signed in as admin	The system will display all the contents	pass	
2.	Search an existing content in search box	Must be signed in as admin	The system will display the existing content	pass	
3.	Search a content that does not exist	Must be signed in as admin	No matching records found	pass	
4.	Search with wrong content name	Must be signed in as admin	No matching records found	pass	

Test Case-13:

Test case Name: Add question

Short Description: Test of adding question

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Test with empty question, empty option and empty answer, then click add	Must be signed in as admin	The system will display “Please fill out this field”	pass	
2.	Test with empty question, option given and answer selected, then click add	Must be signed in as admin	The system will display “Please fill out this field”	pass	
3.	Test with question added, empty option and empty answer, then click add	Must be signed in as admin	The system will display “Please fill out this field”	pass	
4.	Test with question added, option given and empty answer, then click add	Must be signed in as admin	The system will display “Please fill out this field”	pass	
5.	Test with question added, empty option and answer selected, then click add	Must be signed in as admin	The system will display “Please fill out this field”	pass	
6.	Test with empty question, empty option and answer selected, then click add	Must be signed in as admin	The system will display “Please fill out this field”	pass	
7.	Test with empty question, option given and empty answer, then click add	Must be signed in as admin	The system will display “Please fill out this field”	pass	

8.	Test with question added, option given and answer selected, then click add	Must be signed in as admin	The system will display "successfully added question and answer"	pass	

Test Case-14:

Test case Name: Submit answer

Short Description: Test of submitting answers

Test steps	Action	Pre-condition	Expected system response	Pass/Fail	Comments
1.	Select all the answers and click submit button	Must be a registered user	The system will display the obtained marks	pass	
2.	Select no answer and click submit button	Must be a registered user	The system will display the obtained marks	pass	
3.	Select some answers and click submit button	Must be a registered user	The system will display the obtained marks	pass	
4.	Select an answer and refresh the page	Must be a registered user	The system will preserve the selected answer	pass	
5.	Select some answers and click back	Must be a registered user	Redirect to the previously visited page	pass	

Test Case-15:

Test case Name: Add schedule quiz

Short Description: Test of adding schedule quiz

Test steps	**Action**	**Pre-condition**	**Expected system response**	**Pass/Fail**	**Comments**
1.	Test with empty quiz name and date selected, then click add button	Must be signed in as admin	The system will display “Please fill out this field”	pass	
2.	Test with given quiz name and date selected, then click add button	Must be signed in as admin	The system will display “successfully added quiz” and a mail will be sent to each registered user	pass	

Test Case-16:

Test case Name: Sign out

Short Description: Test of signing out

Test steps	**Action**	**Pre-condition**	**Expected system response**	**Pass/Fail**	**Comments**
1.	Test with clicking sign out button	Must be signed in	The system will redirect to the home page	pass	

Chapter 12

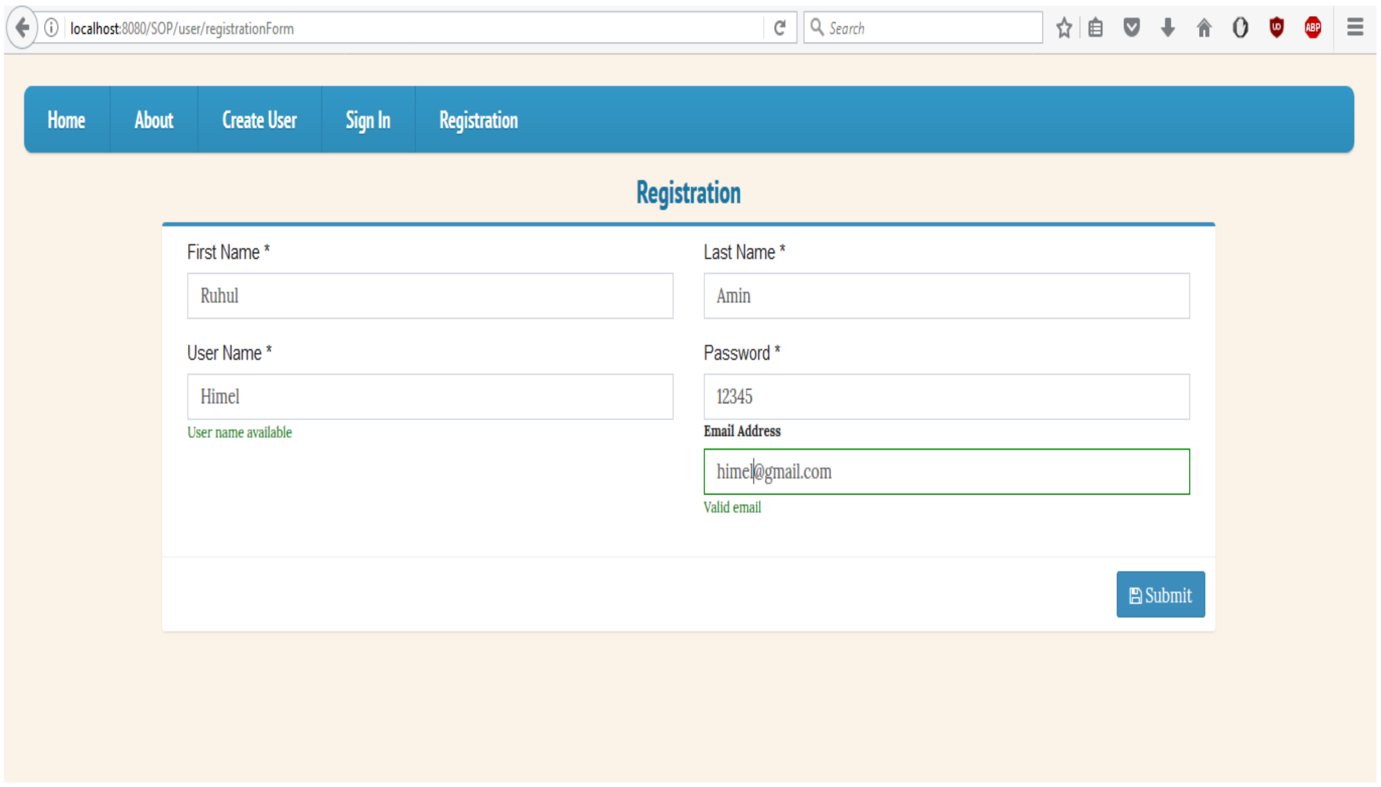
User Manual

12.1 Introduction

A user guide also commonly known as a user manual, is a technical communication document intended to give assistance to people using a particular system. It is usually written by a technical writer. Although user guides are written by programmers, product or project managers, or other technical staff particularly in smaller companies. User guides are most commonly associated with electronic goods, computer hardware and software. Most user guides contain both a written guide and the associated images. In the case of computer applications, it is customary to include screenshots of the human-machine interface.

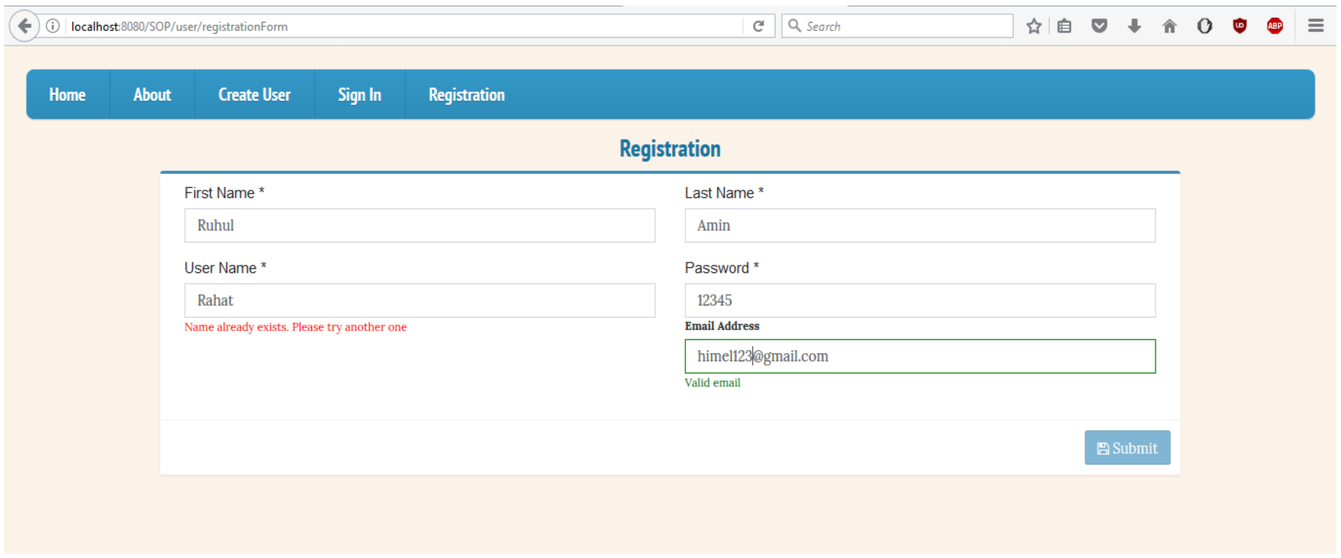
Registration:

This is the registration page. If a user does not have an account, then he/she must do registration to create an account. The user must provide a unique user name & valid email address.



First Name * Ruhul
Last Name * Amin
User Name * Himel
User name available
Password * 12345
Email Address himel@gmail.com
Valid email

If a user does not provide unique user name, an error message will be shown in the following way:



Registration

First Name * Ruhul

Last Name * Amin

User Name * Rahat

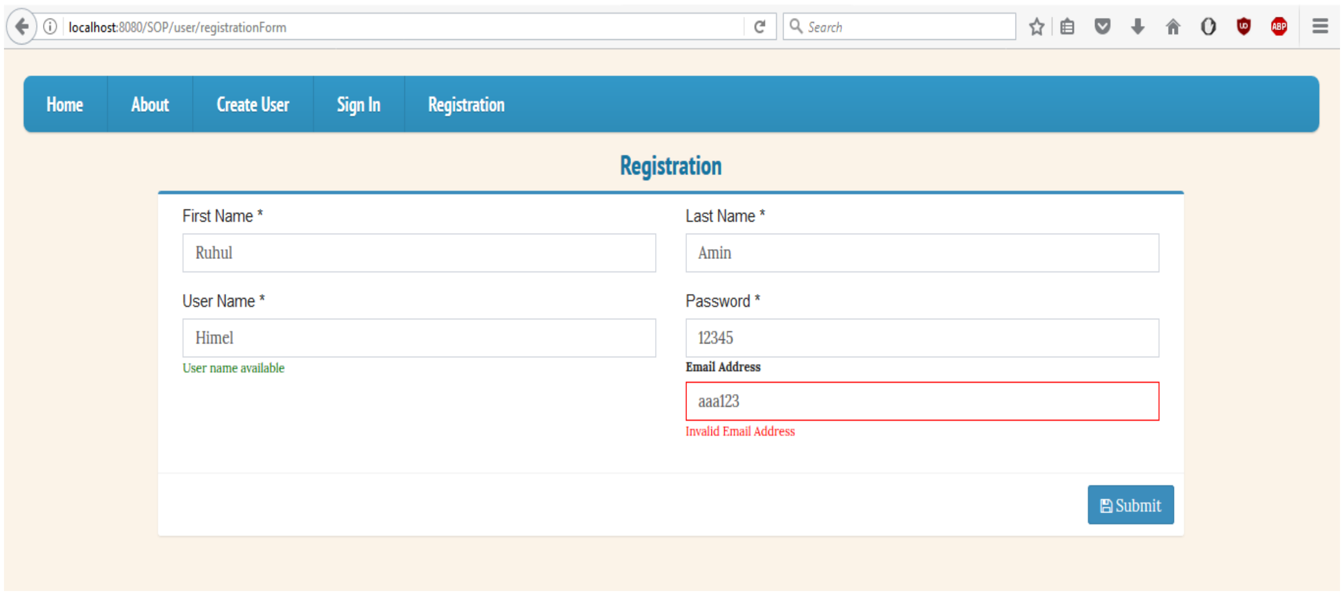
Name already exists. Please try another one

Password * 12345

Email Address himel123@gmail.com

Valid email

If a user does not provide any valid email, an error message will be shown in the following way:



Registration

First Name * Ruhul

Last Name * Amin

User Name * Himel

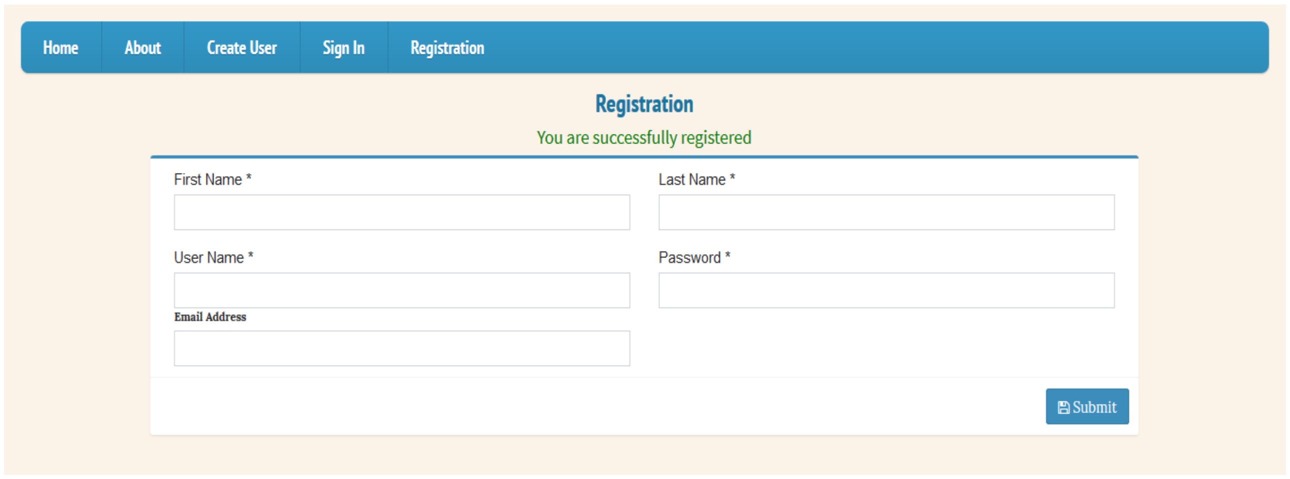
User name available

Password * 12345

Email Address aaa123

Invalid Email Address

After a user will successfully complete registration process, a success message will be shown in the following way:



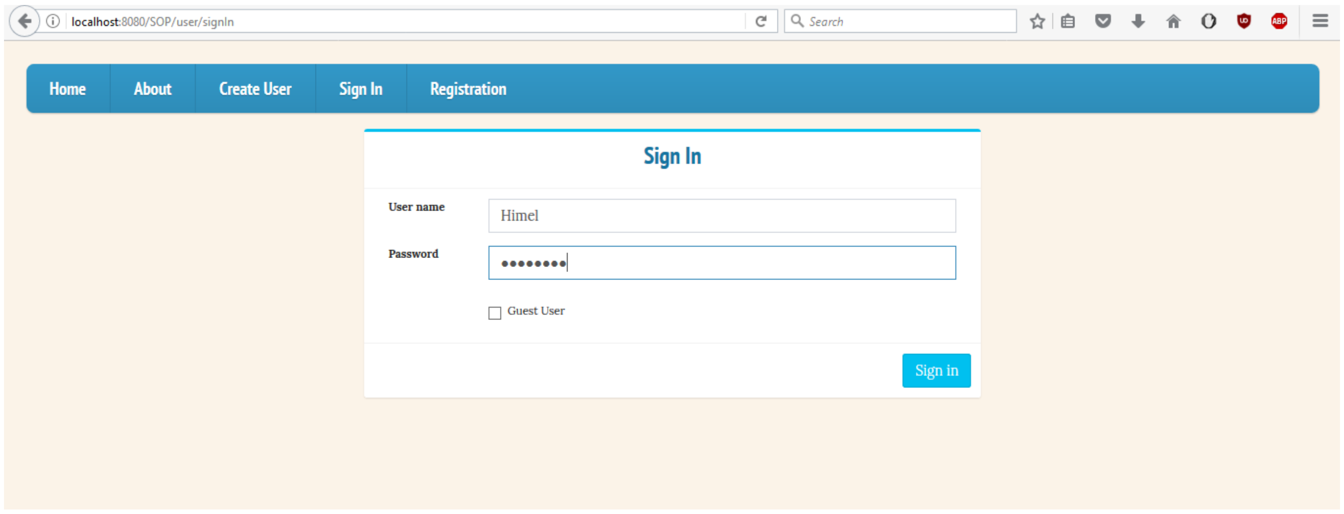
Registration

You are successfully registered

First Name *	Last Name *
User Name *	Password *
Email Address	

Sign in:

A user can sign in the system by providing his user name & password:

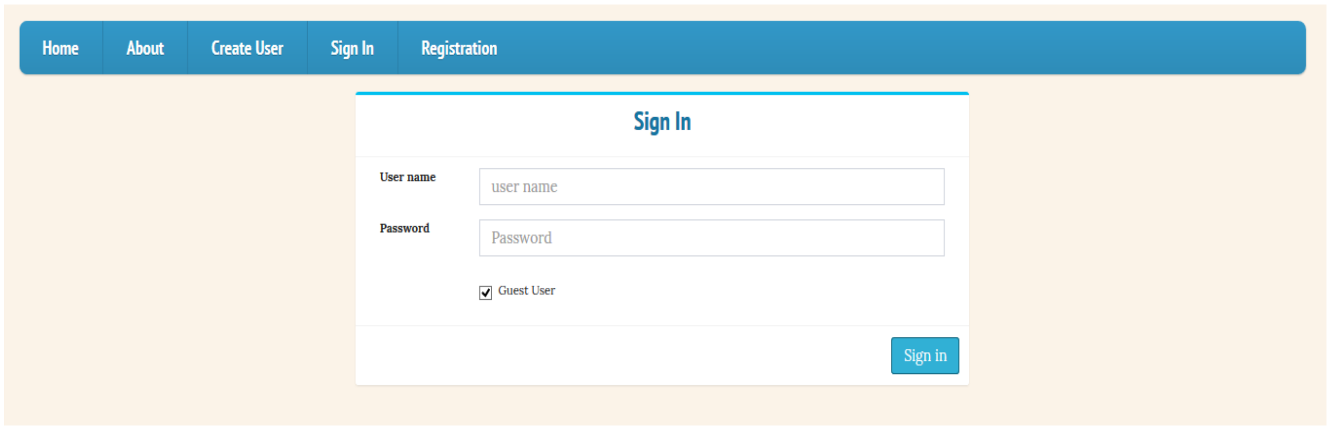


Sign In

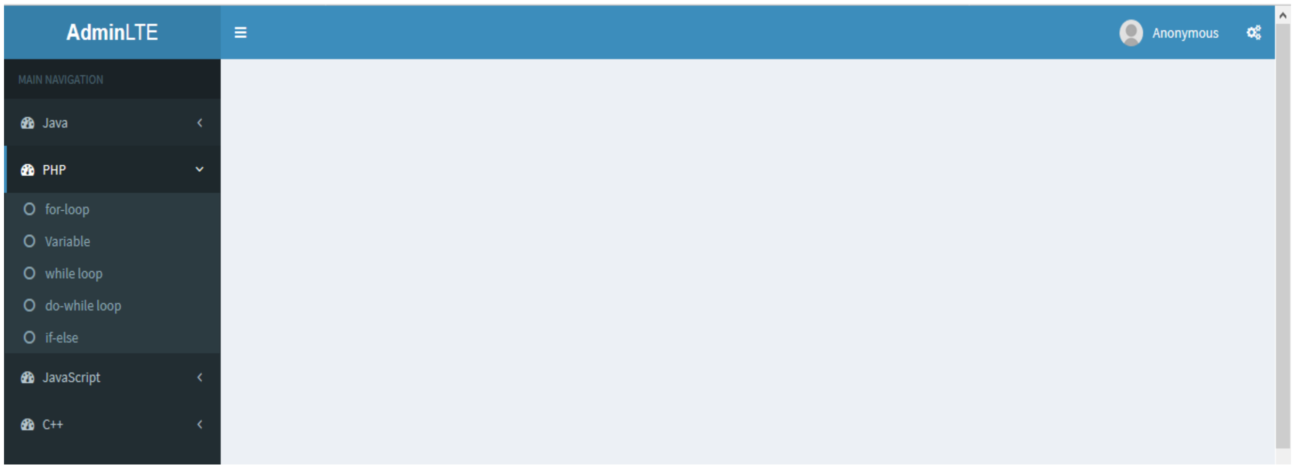
User name	Himel
Password	••••••••

☐ Guest User

If a user is not registered, then he/she is considered as a guest user. A guest user can enter in the system (by selecting 'Guest User' box) in the following way:



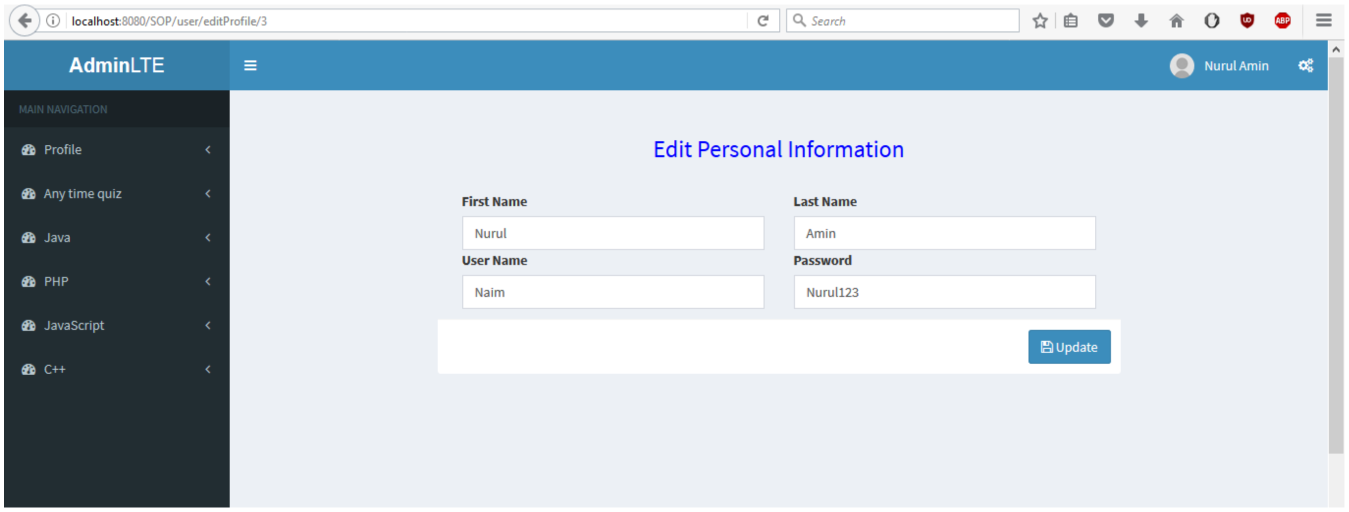
After signing in, a guest user can see the home page in the following way:



A guest user can only go through the content, but he/she cannot attend in the quiz.

Edit profile:

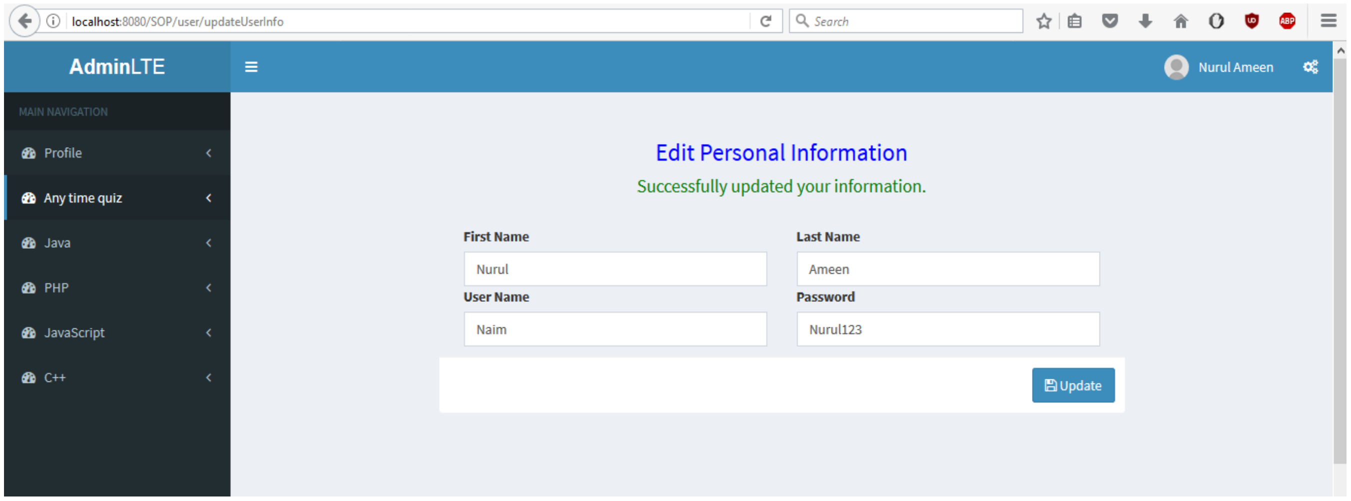
If a user wants to change any personal information, he/she can click **edit profile** and then change information in the following way:



Edit Personal Information

First Name: Nurul
Last Name: Amin
User Name: Naim
Password: Nurul123

After clicking the update button, a message will be shown in the following way:



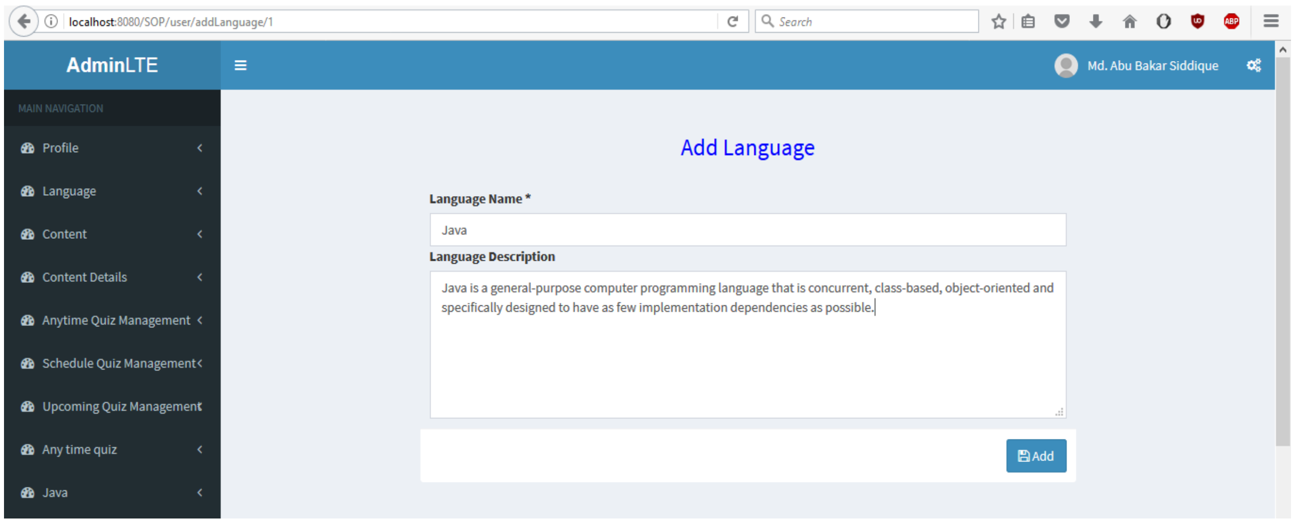
Edit Personal Information

Successfully updated your information.

First Name: Nurul
Last Name: Ameen
User Name: Naim
Password: Nurul123

Add language:

Only the admin of this system can add, edit, search or delete any language. The admin can add language in the system in the following way:



Language Name *

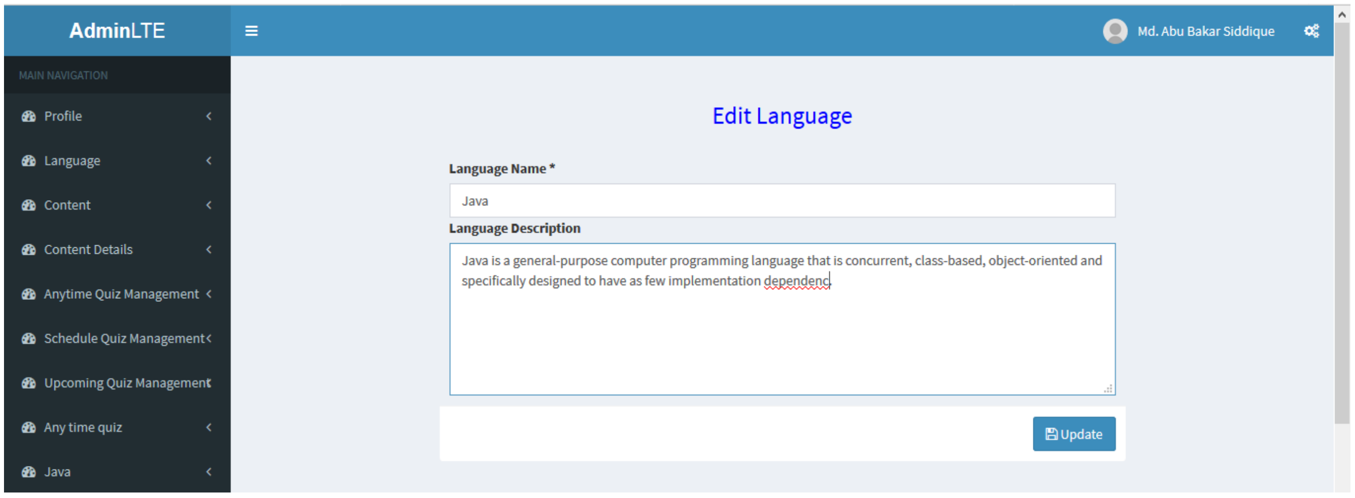
Java

Language Description

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented and specifically designed to have as few implementation dependencies as possible.

Edit language:

If the admin wants to change language information, then he/she can edit the description and click the **update** button:



Language Name *

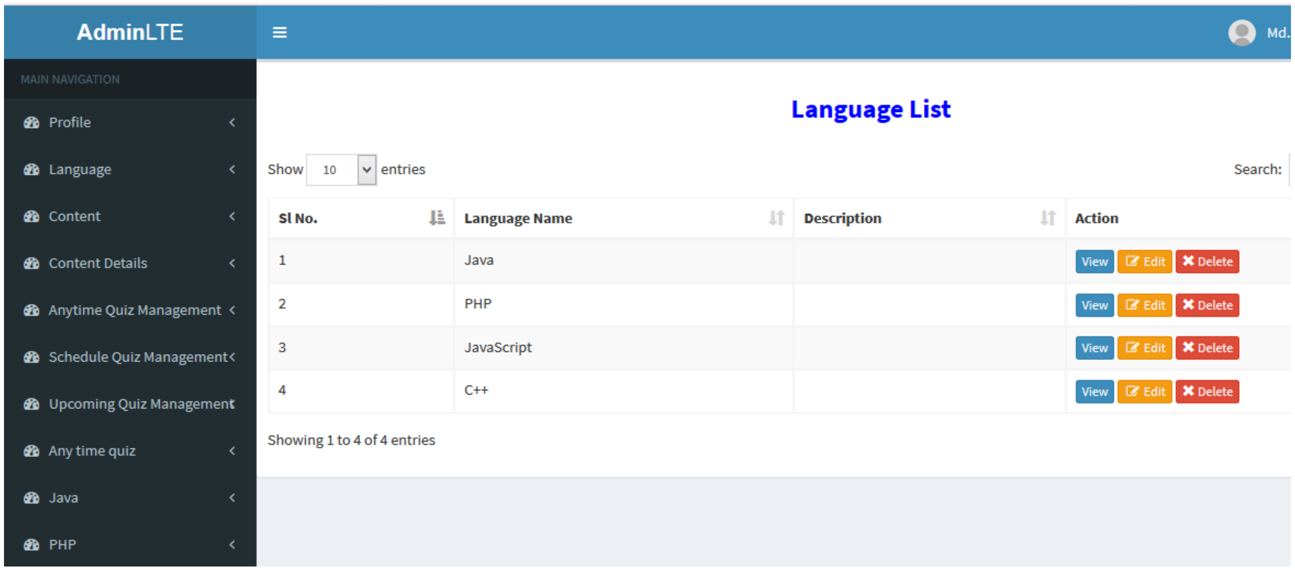
Java

Language Description

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented and specifically designed to have as few implementation dependenc

Delete language:

If the admin wants to delete any language, then he/she needs to click the **delete** button which is marked in ‘red’ color.



Language List

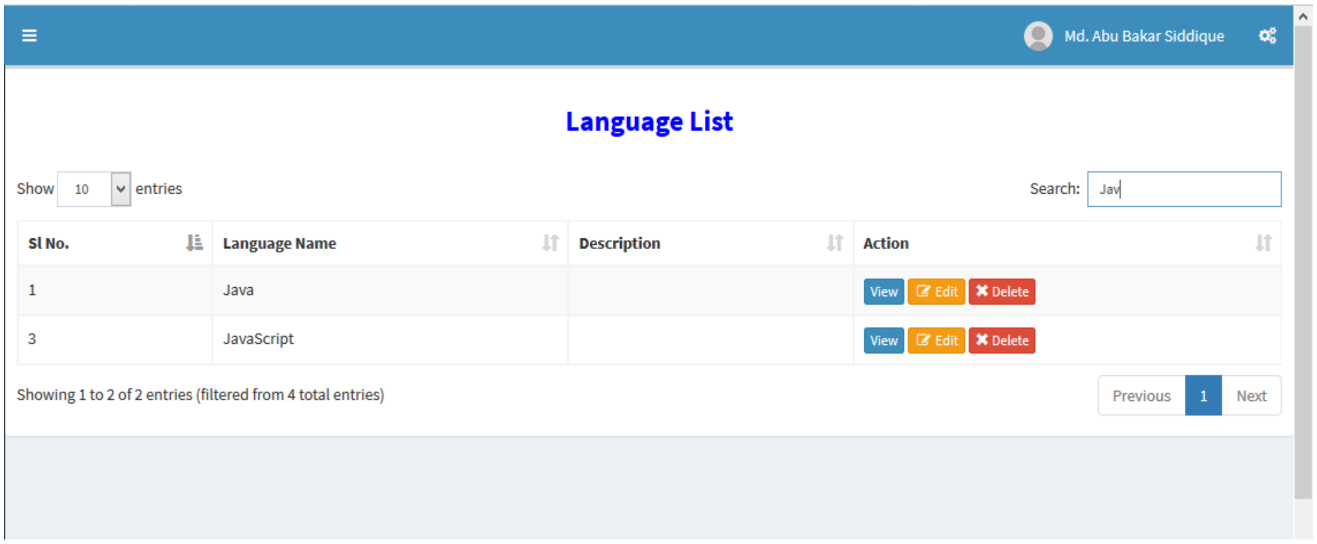
Show 10 entries Search:

Sl No.	Language Name	Description	Action
1	Java		View Edit Delete
2	PHP		View Edit Delete
3	JavaScript		View Edit Delete
4	C++		View Edit Delete

Showing 1 to 4 of 4 entries

Search language:

If the admin wants to search any particular language, then he/she needs to type the desired language name in the search box and see something like this:



Language List

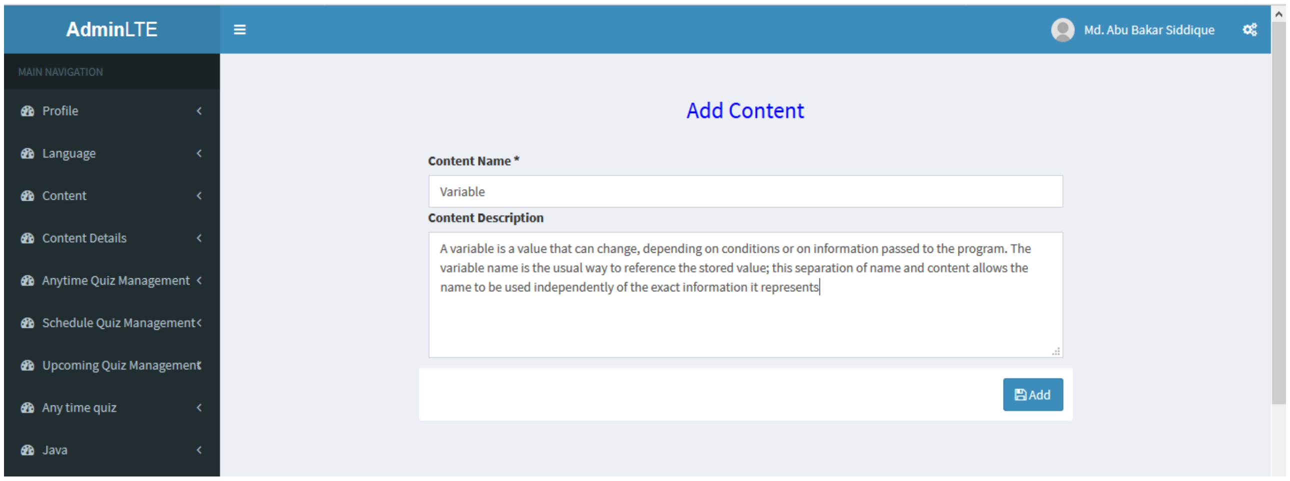
Show 10 entries Search: Jav

Sl No.	Language Name	Description	Action
1	Java		View Edit Delete
3	JavaScript		View Edit Delete

Showing 1 to 2 of 2 entries (filtered from 4 total entries)

Add Content:

Only the admin of this system can add, edit, search or delete any content. The admin can add content in the system in the following way:



Add Content

Content Name *

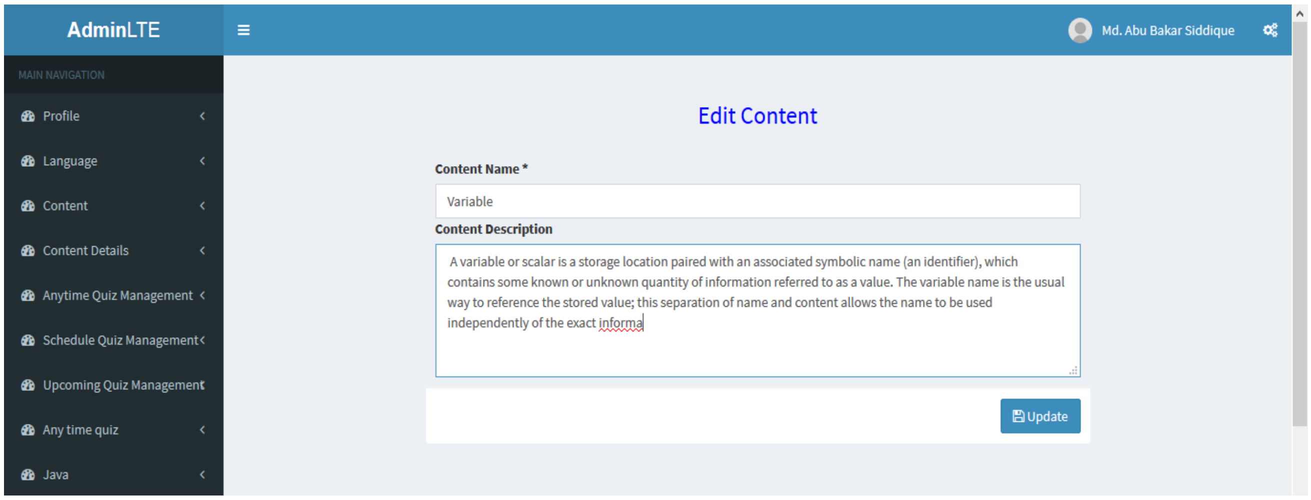
Variable

Content Description

A variable is a value that can change, depending on conditions or on information passed to the program. The variable name is the usual way to reference the stored value; this separation of name and content allows the name to be used independently of the exact information it represents

Edit Content:

If the admin wants to change content information, then he/she can edit the description and click the **update** button:



Edit Content

Content Name *

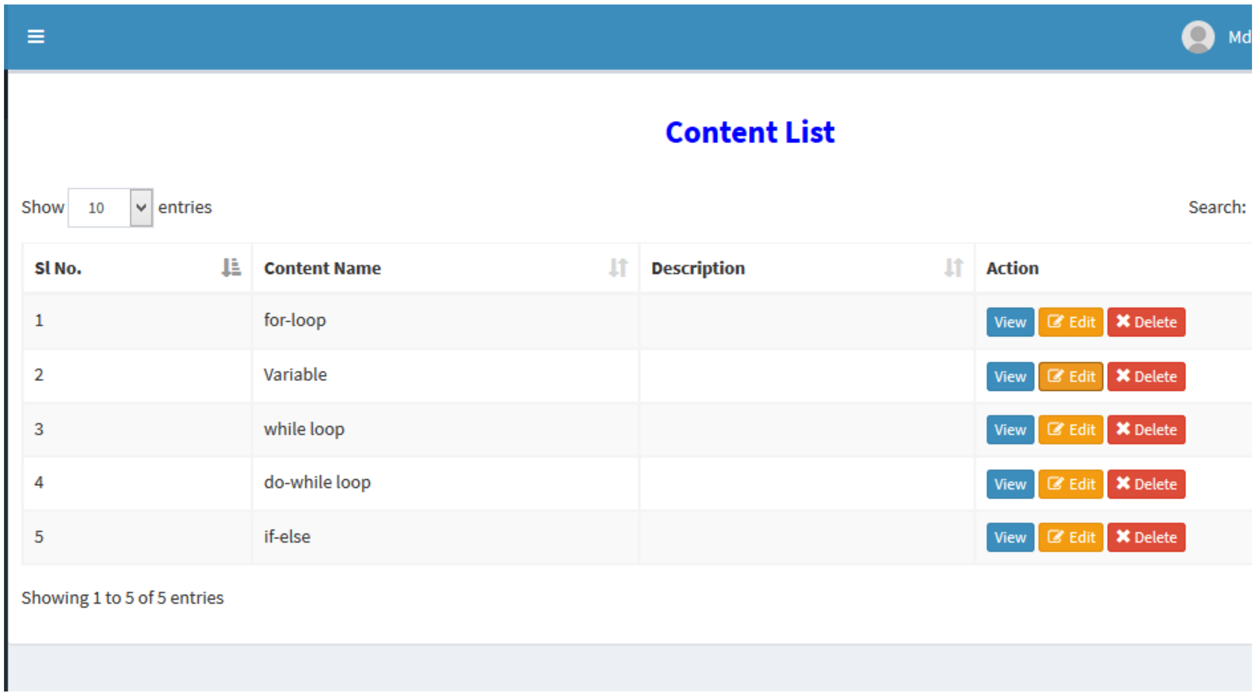
Variable

Content Description

A variable or scalar is a storage location paired with an associated symbolic name (an identifier), which contains some known or unknown quantity of information referred to as a value. The variable name is the usual way to reference the stored value; this separation of name and content allows the name to be used independently of the exact informa

Delete Content:

If the admin wants to delete any content, then he/she needs to click the **delete** button which is marked in 'red' color.



Md

Content List

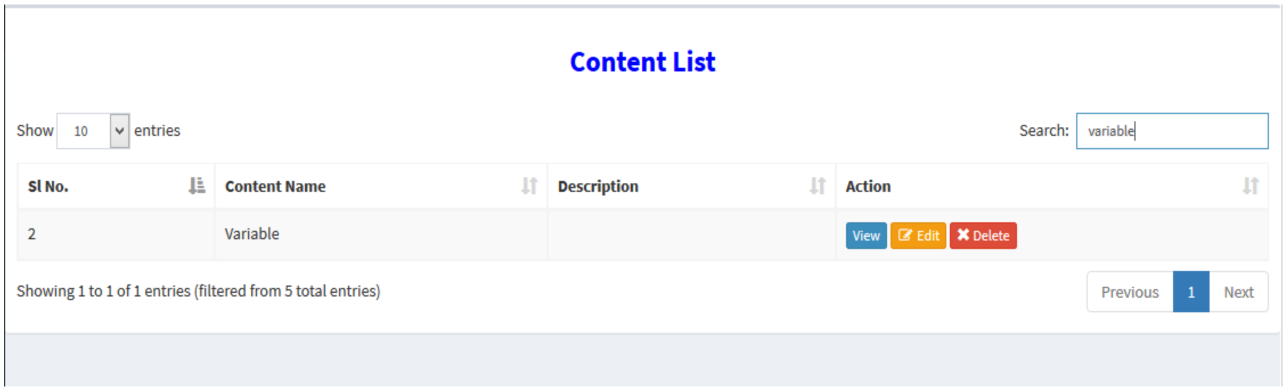
Show 10 entries Search:

Sl No.	Content Name	Description	Action
1	for-loop		View Edit Delete
2	Variable		View Edit Delete
3	while loop		View Edit Delete
4	do-while loop		View Edit Delete
5	if-else		View Edit Delete

Showing 1 to 5 of 5 entries

Search language:

If the admin wants to search any particular language, then he/she needs to type the desired language name in the search box and see something like this:



Content List

Show 10 entries Search: variable

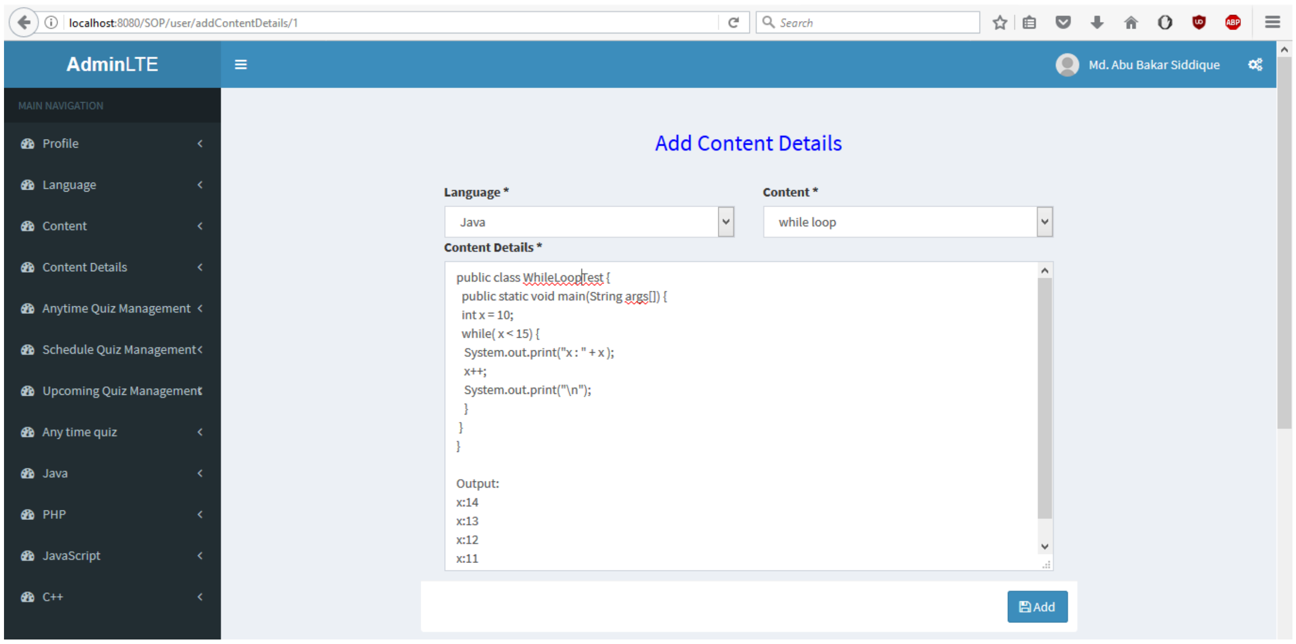
Sl No.	Content Name	Description	Action
2	Variable		View Edit Delete

Showing 1 to 1 of 1 entries (filtered from 5 total entries)

Previous 1 Next

Add Content details:

Only the admin of this system can add, edit, search or delete any content details. The admin can add content details in the system in the following way:



Add Content Details

Language * Java

Content * while loop

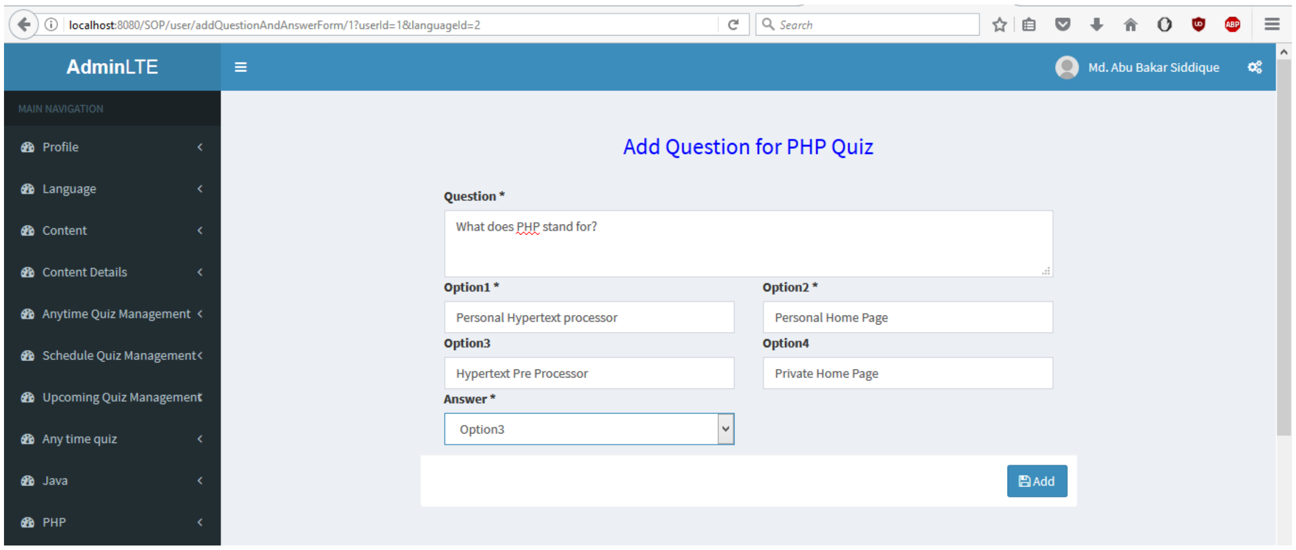
Content Details *

```
public class WhileLoopTest {
 public static void main(String args[]) {
 int x = 10;
 while( x < 15) {
  System.out.print("x : " + x );
  x++;
  System.out.print("\n");
  }
 }
}

Output:
x:14
x:13
x:12
x:11
```

Add Questions for quiz:

The admin adds question for both the anytime quiz and schedule quiz in the following way:



Add Question for PHP Quiz

Question * What does PHP stand for?

Option1 * Personal Hypertext processor

Option2 * Personal Home Page

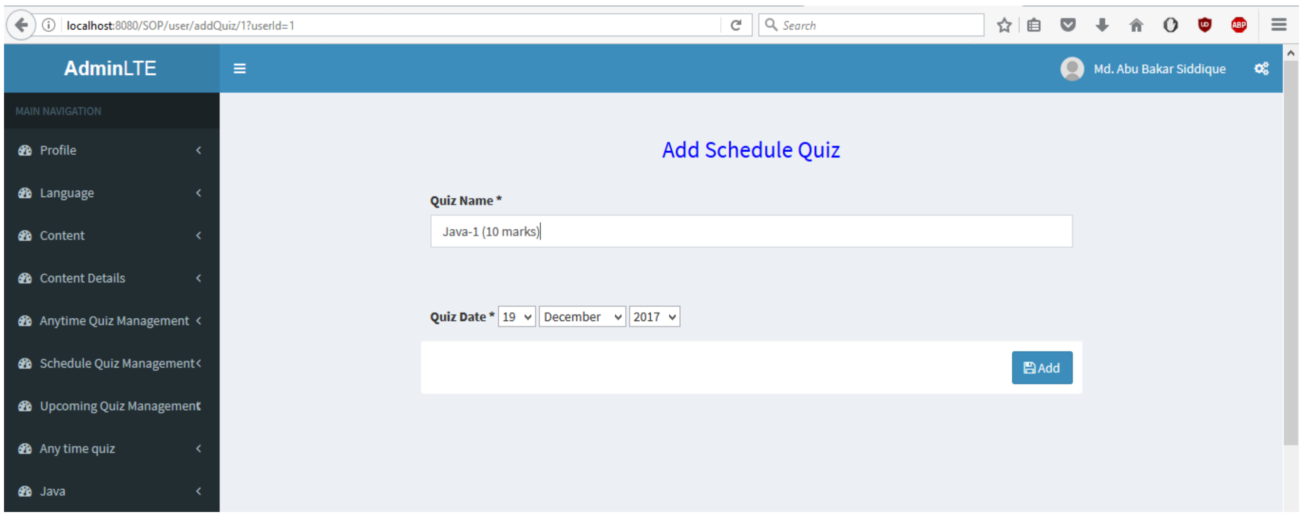
Option3 Hypertext Pre Processor

Option4 Private Home Page

Answer * Option3

Add Schedule quiz:

The admin decides a date and particular language for taking schedule quiz. Then he notifies all the registered user.



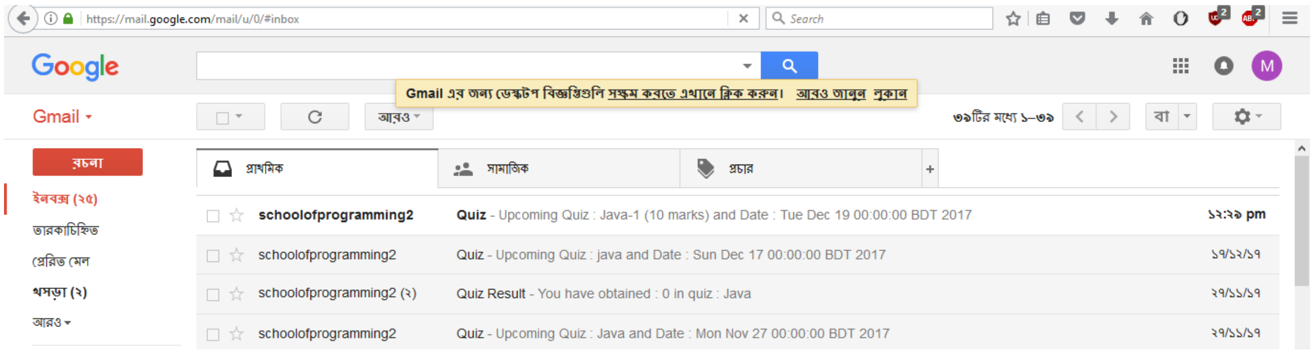
Add Schedule Quiz

Quiz Name * Java-1 (10 marks)

Quiz Date * 19 December 2017

Receive mail:

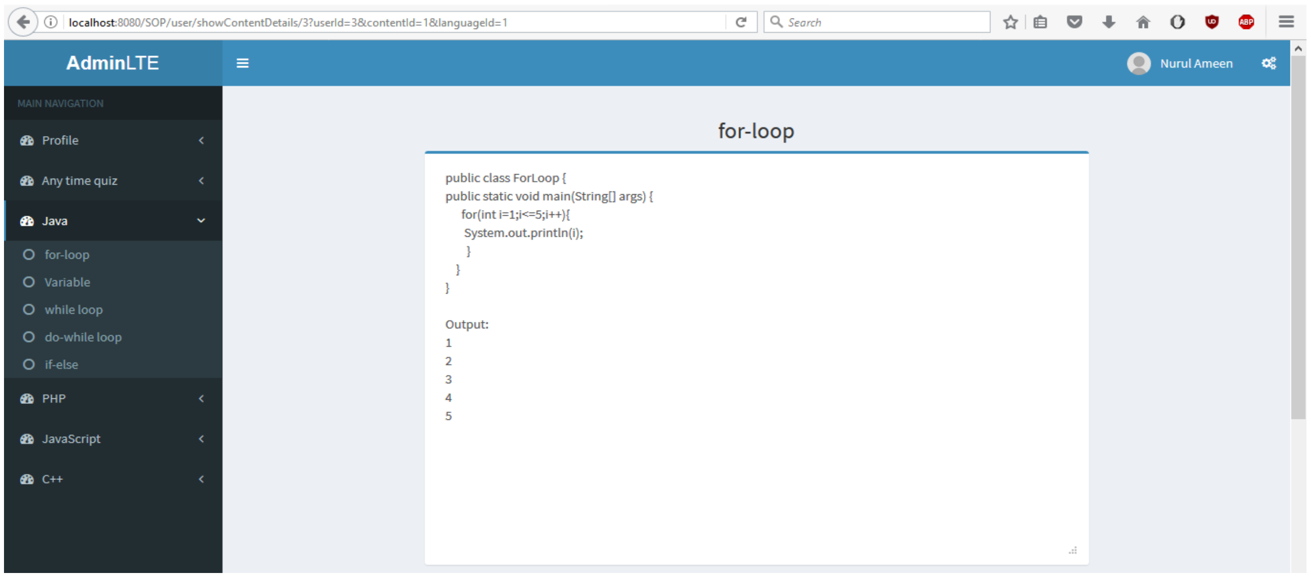
All the register user will get a mail as soon as the admin declares the schedule quiz date and syllabus. A mail interface will look like this. The mail is sent from the system ‘School of programming’.


https://mail.google.com/mail/u/0/#inbox
Google
Gmail
Gmail এর জন্য ডেস্কটপ বিজ্ঞপ্তিগুলি সক্ষম করতে এখানে ক্লিক করুন। আরও জানুন লুকান
আরও
৩৯টির মধ্যে ১–৩৯
বা
রচনা
ইনবক্স (২৫)
তারকাচিহ্নিত
প্রেরিত মেল
খসড়া (২)
আরও
প্রাথমিক
সামাজিক
প্রচার

schoolofprogramming2	Quiz - Upcoming Quiz : Java-1 (10 marks) and Date : Tue Dec 19 00:00:00 BDT 2017	১২:২৯ pm
schoolofprogramming2	Quiz - Upcoming Quiz : java and Date : Sun Dec 17 00:00:00 BDT 2017	১৭/১২/১৭
schoolofprogramming2 (২)	Quiz Result - You have obtained : 0 in quiz : Java	২৭/১১/১৭
schoolofprogramming2	Quiz - Upcoming Quiz : Java and Date : Mon Nov 27 00:00:00 BDT 2017	২৭/১১/১৭

View details:

A general user and guest user can view the contents in the following way:



localhost:8080/SOP/user/showContentDetails/3?userId=3&contentId=1&languageId=1

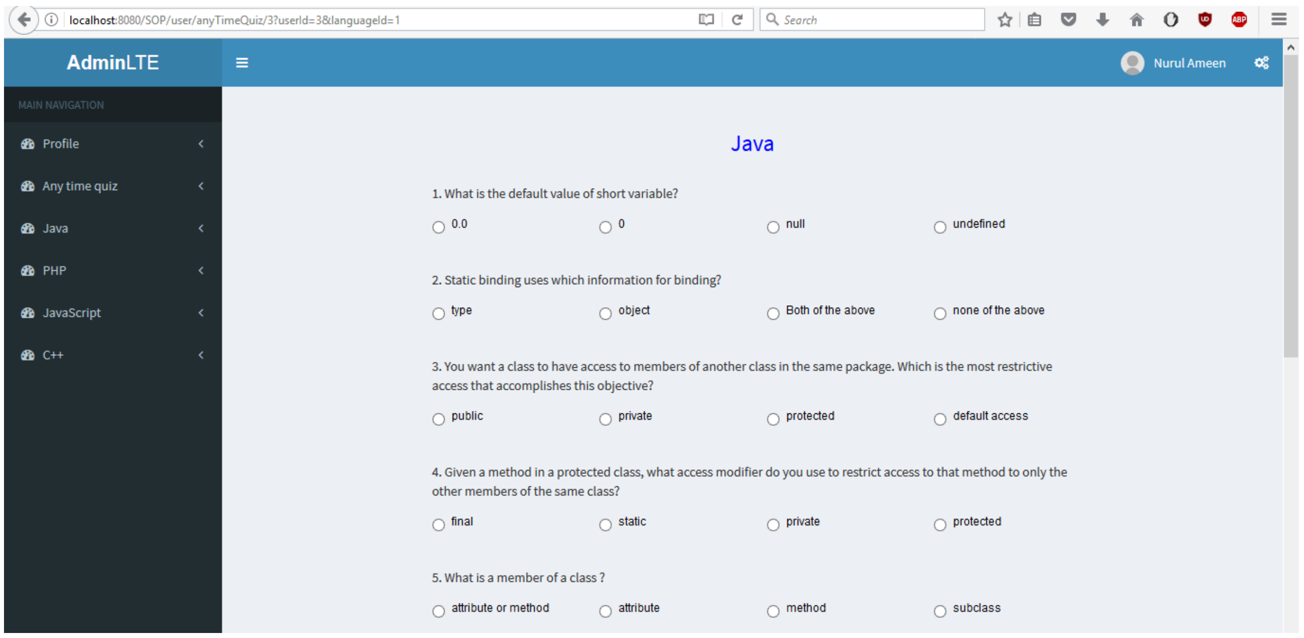
for-loop

```
public class ForLoop {
public static void main(String[] args) {
    for(int i=1;i<=5;i++){
     System.out.println(i);
     }
   }
}

Output:
1
2
3
4
5
```

Participate in quiz:

Only a registered user can participate in the quizzes. The quiz interface will appear as following:



localhost:8080/SOP/user/anyTimeQuiz/3?userId=3&languageId=1

Java

1. What is the default value of short variable?

○ 0.0 ○ 0 ○ null ○ undefined

2. Static binding uses which information for binding?

○ type ○ object ○ Both of the above ○ none of the above

3. You want a class to have access to members of another class in the same package. Which is the most restrictive access that accomplishes this objective?

○ public ○ private ○ protected ○ default access

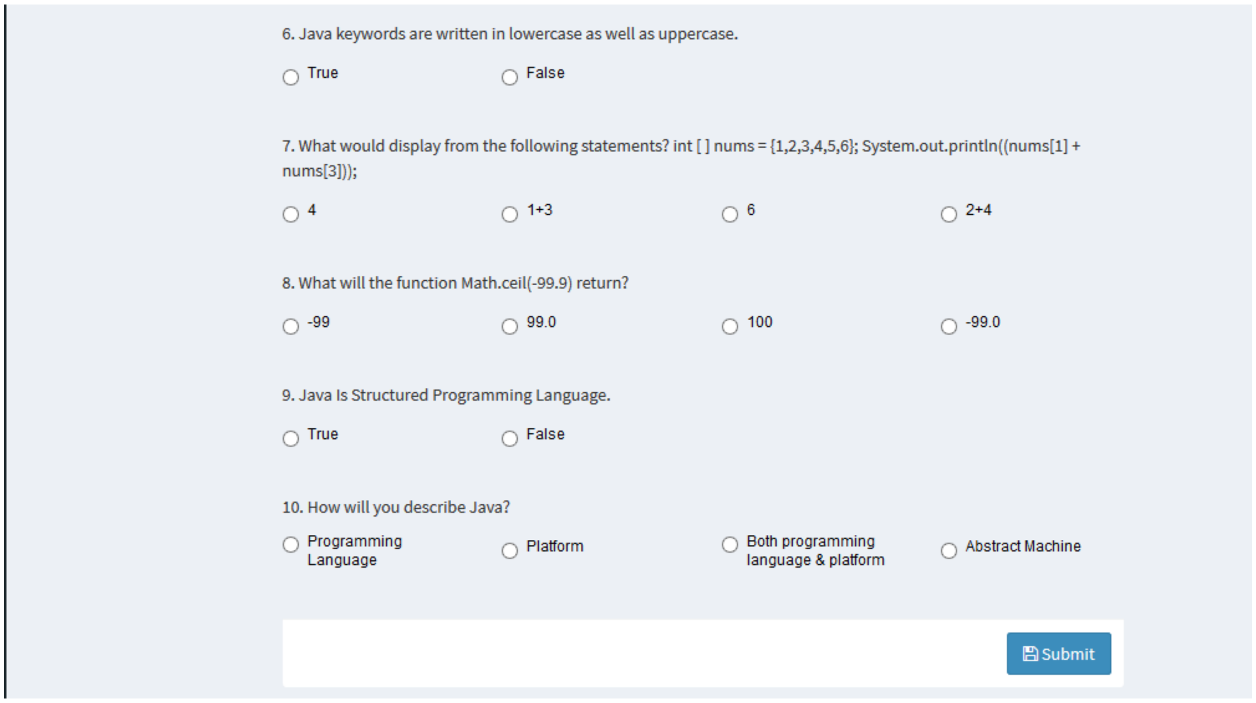
4. Given a method in a protected class, what access modifier do you use to restrict access to that method to only the other members of the same class?

○ final ○ static ○ private ○ protected

5. What is a member of a class ?

○ attribute or method ○ attribute ○ method ○ subclass

The user can discontinue the exam at any time and finish the quiz by clicking the **submit** button:



6. Java keywords are written in lowercase as well as uppercase.

○ True ○ False

7. What would display from the following statements? int [ ] nums = {1,2,3,4,5,6}; System.out.println((nums[1] + nums[3]));

○ 4 ○ 1+3 ○ 6 ○ 2+4

8. What will the function Math.ceil(-99.9) return?

○ -99 ○ 99.0 ○ 100 ○ -99.0

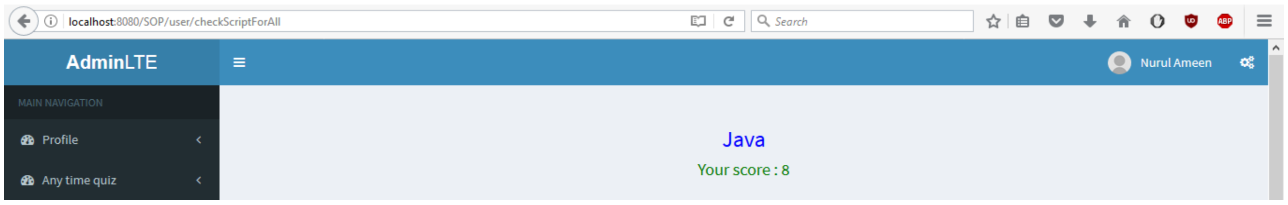
9. Java Is Structured Programming Language.

○ True ○ False

10. How will you describe Java?

○ Programming Language ○ Platform ○ Both programming language & platform ○ Abstract Machine

The user can see the marks of any time quiz instantly in the following way:



Java

Your score : 8

Reference

1. Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th ed.). Boston, Mass: McGraw-Hill. ISBN 0-07-285318-2
2. Database System Concepts, 5th Ed. ©Silberschatz, Korth and Sudarshan
3. Sommerville, I. Software Engineering, 7th ed. Harlow, UK: Addison Wesley, 2006