College of Natural and Computational Sciences

Department of Computer Science

Requirement Analysis Document Draft 1

Project: AceSAT – SAT Prep Software write the full name of the system

Group Members:

Lydia Yoseph	UGR/7809/15
Meseret Ghebiresilassie	UGR/0722/15
Salem Gebru	UGR/6916/15

Submitted to: Instructor Ayalew Belay (PhD)

Date of Submission: 12/09/2024

Table of Contents

1 Introduction

This document is prepared to record and organize the work carried out during the AceSAT project. It serves as a comprehensive reference for planning, analysis, and design stages, reflecting the methodologies and practices learned in our Software Engineering course. The primary goal of this project is not only to deliver a functional SAT preparation system but also to enhance our understanding of software engineering principles and gain hands-on experience in applying them to real-world scenarios.

By addressing the challenges Ethiopian students face in SAT preparation, the project aims to provide personalized practice questions, mock exams, and study plans, improving accessibility to global education opportunities.

1.1 Purpose of the Document

The purpose of this document is to provide a structured framework for the analysis, design, and development of AceSAT. It outlines the system requirements and serves as a roadmap for the project's implementation, ensuring alignment with stakeholder needs and educational objectives. Additionally, it reflects the application of software engineering principles in a practical, real-world context.

1.2 Scope of the Project

- **Target Audience:** Students in high school or university
- **Geography:** Focused on Ethiopian students
- **Subject Coverage:** Math and English
- **Features:** The initial development phase will focus on delivering core functionalities to provide a seamless and user-friendly experience. These core features include:
  - User Registration: Allowing new users to create accounts easily.
  - User Account Authentication: Ensuring secure login and account management.

Additional features are planned for implementation in future sprints to enhance the system's functionality and user experience. These features include, but are not limited to:

- Practice Sets: Offering curated SAT practice questions.
- Timed and Untimed Practice Sessions: Allowing students to customize practice based on their preferences.
- Mock Examinations: Providing full-length SAT simulations with realistic time limits.

    - Explanations for Missed Questions: Helping users learn from their mistakes by offering detailed explanations.
    - Progress Tracking: Enabling students to monitor their performance over time.
    - Comprehensive SAT Information: Offering detailed guidance about the SAT, including registration steps, test structure, and scoring.
- **Technical scope:** Web application

1.3 Document Overview

This Requirements Analysis Document is the initial draft in a series of evolving documents that will be updated as new features are added to the system and the requirements are refined. This version is structured to outline the current features and provide a comprehensive view of the project's foundation.

The document includes the following sections:

- **Introduction**: Purpose of the document, scope of the project, definitions, acronyms, abbreviations, and references.
- **Project Overview**: Background, objectives, success criteria, scope and boundaries.
- **Stakeholder Analysis**: Identification of key stakeholders and their roles.
- **Requirements Elicitation**: Techniques used, scenarios, and use cases.
- **Current System Analysis**: Overview of the existing system or context.
- **Requirements Analysis and Specification**: Detailed functional and non-functional requirements.
- **System Model**: Visual and descriptive representations of the system.

This document serves as a foundational reference, guiding the development process and ensuring alignment with stakeholder needs and project goals.

1.4 Definitions, Abbreviations, and Acronyms

- **SAT:** Scholastic Aptitude Test
- **UC:** Use Case
- **Actor/User:** Students in high school or university

1.5 References

- Das, S. (2024, November 14). *How to write agile software requirement specifications | BrowserStack*. BrowserStack. https://www.browserstack.com/guide/software-requirement-specifications-in-agile

- Snappify. (n.d.). *Software Requirements Specification Sample: 101 Expert Guide*. https://snappify.com/blog/software-requirements-specification-sample

2 Project Overview

2.1 Project Background and Development Model

The Scholastic Assessment Test (SAT) is a globally recognized standardized test widely used for college admissions, particularly in the United States. Ethiopian students face barriers such as limited access to practice resources, lack of proper registration guidance, and difficulties in understanding the test format. AceSAT aims to address these challenges by offering curated and accessible SAT preparation tools.

The chosen development model for this project is the Agile Development Model. This approach is preferred due to the challenges involved in addressing all project requirements at once. Given that our end users are likely to have evolving and diverse functional and design needs, the Agile methodology allows for flexibility and responsiveness. By gathering requirements incrementally and iteratively, we can ensure that the project aligns with user expectations over time. Each sprint will deliver a set of features, allowing for continuous feedback and improvements, which helps us adapt quickly to changes in requirements or priorities.

2.2 Objective and Success Criteria

the objective is to develop S.... which will have benefits including

The project intends to resolve the issues students encounter when trying to prepare for the international SAT examinations. It does this by giving them a general overview of the exam, personalized practice questions**,** and aiding their study process by keeping track of their progress.

The project shall be successful if it maintains students' privacy, allows access to the exam banks and ultimately improves the student’s knowledge of the exam and their preparedness.

2.3 Scope and Boundaries

The system’s scope includes features like user registration, personalized practice, mock exams, and progress tracking. Boundaries exclude providing live tutoring or SAT registration services directly.

2.4 Assumptions and Dependencies

- Students have access to internet-enabled devices.
- The system relies on accurate SAT preparation materials.
- Development depends on regular feedback from stakeholders.

3 Stakeholder Analysis

3.1 Stakeholder Identification

- **Students**: Primary end users who will use the system for SAT preparation.
- **Project Team**: Developers and designers of the AceSAT system.

3.2 Roles and Responsibilities

- **Students**: Provide feedback on system usability and content quality.
- **Project Team**: Implement, test, and deploy the system while incorporating feedback.

4 Requirements Elicitation

4.1 Elicitation Methods

- **Surveys and Questionnaires**: To gather broad input from the target audience.
- **Interviews**: To gain in-depth understanding of user needs and challenges.
- **Observation**: To assess existing SAT preparation practices.

4.2 Documented Scenarios and Use Cases

- Scenario: A user opens the app for the first time and wants to create an account to get started with their SAT prep.
  - Use case: registerUser and authenticateUser
- Scenario: The user wants to edit his/her profile and set goals so they can track themselves overtime.
  - Use case: editProfile

5 Current System

Ethiopian students intending to take SAT examinations usually have to search for examinations from a variety of sources thoroughly. This can be time-consuming and students may get frustrated in allocating the right resource for their practice sessions. In addition, they can't find detailed information about what the registration and examination process in Ethiopia looks like.

Problems

- Most students do not have access to SAT prep resources like textbooks, or exam questions with proper explanations of the answers.
- Most students do not know the proper registration methods and generally do not know how the exam is administered.
- Students have a hard time keeping track of their progress and being consistent with their studies.

6 Requirement Analysis and Specification

6.1 Functional Requirements

6.1.1 User Registration and Authentication

- Users shall be able to create an account using their email account as their 2-factor authentication.
- Users shall be able to log in and log out securely.
- Password recovery options shall be available, ideally through users' email accounts.

6.1.2 User Profile Management

- Users shall be able to view and edit their profile information (name, email, profile picture).
- Users shall set and update their study goals and preferences.

6.2 Non-Functional Requirements

6.2.1 Security

- User authentication: strong password and email verification
- User data encryption

6.2.2 Usability

- **Minimal Learning Curve:** Easy-to-understand onboarding for first-time users to navigate the app quickly.
- **Responsive Design:** Works well on various devices (smartphones, tablets, and desktops) and screen sizes.

6.2.3 Reliability

- **99.9% Uptime:** The app aims for near-continuous availability, with less than 0.1% downtime (about 8.76 hours annually).
- **Automatic Session Saving:** The app shall automatically save user progress in real-time so that any unexpected exit (e.g., app crash or phone restart) does not lead to data loss.

6.2.4 Performance

- **Fast Loading Times:** The app is expected to load within 2-3 seconds on most devices, including the main dashboard and study content sections.
- **Instant Feedback on Actions:** Users shall receive immediate feedback (less than 1 second) after answering questions or completing actions.
- **Optimized Database Queries:** Database interactions, such as fetching user progress or saving results, shall be optimized for minimal response times.

6.2.5 Interface

- Editing personal information
- Sign Up and Login pages

6.2.6 Legal

- Users shall be asked for consent for any data collection, especially for sensitive data like personal information, location, or usage behavior, and be notified of the purpose of the collected data.
- Encryption and secure data storage shall be used to protect user information.
- This application isn't affiliated with any educational institution.
- The application's exams and quizzes are for educational purposes only and may not guarantee specific exam results.
- Data collected from users include personal information (like email, username, and educational background) and goal-setting data. This data will be used for the creation of personalized accounts and tracking of the student's progress.

7 Proposed System

7.1 Overview

This project is an SAT preparation software for students who are about to take the internationally administered Scholastic Aptitude Test. It is expected to help the students study by giving them practice questions and certified mock examinations. It shall also help them keep track of their progress by giving them personalized study plans. In addition, this project is planned to provide the students general information about the SAT exam.

7.2 User Personas

	User 1	**User 2**
Name	Amir Ahmed	Mahlet Asmamaw
Age	19	17
Nationality	Ethiopian	Ethiopian
Background	University student	Highschool student
Goals	Take the SAT examination to explore abroad education options	Take the SAT examination to explore abroad education options
Challenges	Lacks information on how to register or take the examination here in Ethiopia	Lack of a resource bundle to help her prepare

7.3 User Stories

As a	I want to	So that	Use case
Student	be able to create a personal account	I can have access to personalized content	registerUser
Student	access my account	I can access the content	authenticateUser
Student	edit my personal information and set practice goals	I can have personalized content	editProfile

7.4 System Models

Use case name	Goal	Use case ID
registerUser	Enroll students into the system	UC001
authenticateUser	Authenticate users when logging into the system	UC002
validateInput	Type checking of item values	UC003
editProfile	Set and edit personal information	UC004

7.4.1 Use Case Model

use case model is presented before description

7.4.1.1 Use cases

Use case ID	UC001
Use case name	registerUser
Participating Actors	Student
Entry condition	The registration site is open and accessible to the user.
Flow of events	○ **Student initiates sign up**: The student clicks the option for sign up functionality. ○ **System Displays Required Fields**: The system displays a list of required fields such as username, email, password with relogging of password for confirmation. ○ **Student Edits Information**: The student fills the required information fields and clicks “submit”. ○ **System initiates validation:** The system gets the input type checked and if correct it passes to the next step. ○ **Two step verification**: The system sends an email to the user containing one time OTP code and asks the user to enter the code into the system. ○ **System Saves Changes**: If verification code is sent to email, the system saves the entered information to the database. ○ **OTP Verification:** The system verifies the OTP within the specified time and if it’s valid and entered within the specified time, the system redirects to the login page.
Exit condition	The student has been registered into the system.
Alternative flow	○ If the entered data is in invalid format or data, the system will issue a message to enter in the correct specified format. ○ If the OTP is incorrect, it allows the user to request a resend option, where an OTP is regenerated.

give name to your system instead of calling as system

Use Case ID	UC002
Use case name	authenticateUser
Participating actors	Student
Entry condition	The login page is accessible and the student has valid login credentials.
Flow of events	o **Student initiates log in:** The student selects the “log in” option. o **Student enters required information:** The student enters email and password. o **Information verification:** The system checks the database to verify the entered information. If the credentials are correct, the system allows user entry to the dashboard.
Exit condition	The student is successfully logged in and redirected to the dashboard.
Alternative flow	o If the email isn’t correct, it tells the user “Email is incorrect” and asks the user to relog. o If the password doesn’t match the stored password in the database, it tells the user “Password is incorrect” and asks the user to relog. o If the password is forgotten, the system gives the option to send the password to the student’s email.

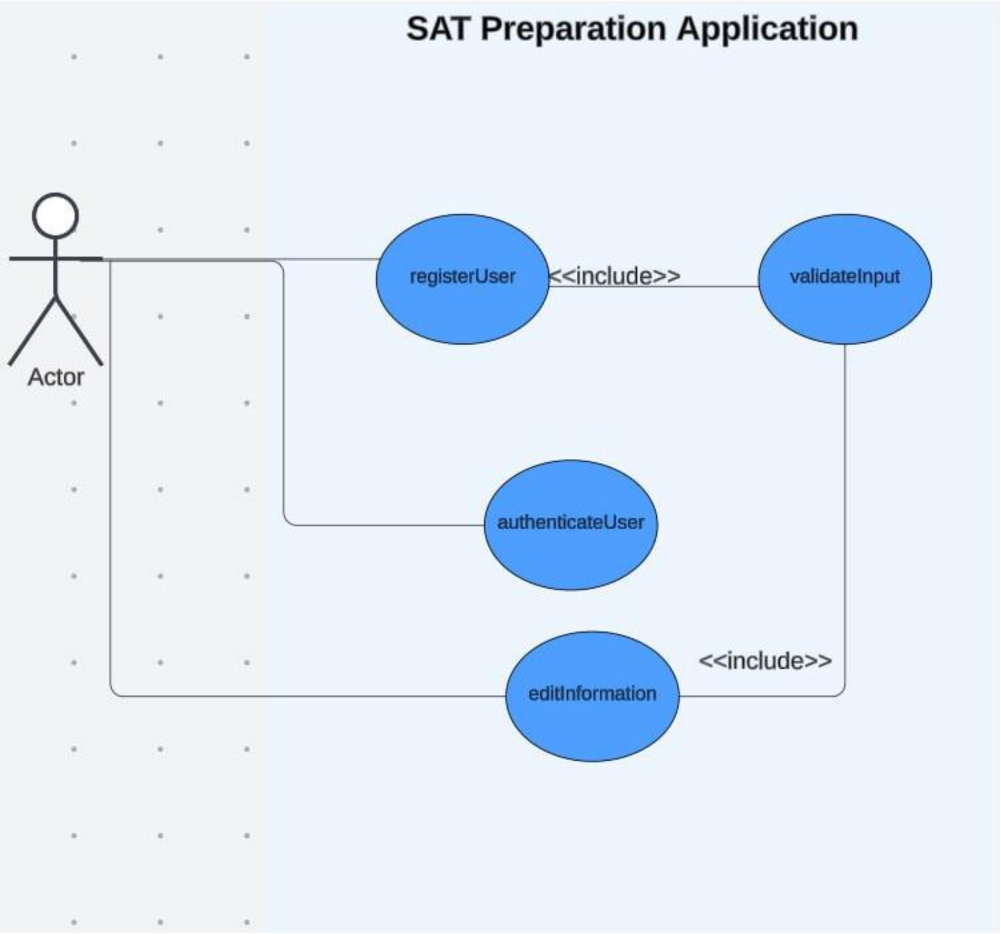
Use case ID	UC003
Use case name	validateInput
Participating Actors	Student
Entry condition	Input fields are available and the system has received input.
Flow of events	o **Student logs fields:** Student enters the required information. o **System typechecks:** The system validates the formatting of email and password according to the standard formula. If the fields pass the check, it transfers the student to the next use case.

Use case ID	UC003
Exit condition	The student credentials have been verified.
Alternative flow	o If a field is empty, the system tells the user to enter that information. o If the field doesn't match the format, the system tells the user to format it accordingly.

Use case ID	UC004
Use case name	editProfile
Participating Actors	Student
Entry condition	The student is authenticated and logged in
Flow of events	o **Student Logs In**: The student logs into their account to access exam content. o **Student navigates to profile section:** The student selects the button for editing profile o **Student fills the required fields:** The student will fill up the editable fields such as full name and target score. Student also sets goals for practice question number and target score. o **System typechecks:** The system validates the formatting of credentials according to the standard formula. If the fields it lets the student move on to the next field. o **Student submits changes:** By clicking save changes, the student commits the alterations to the system. o **System gives confirmation message:** The system gives a message that the changes have been saved.
Exit condition	Information fields are updated and logged into the student's profile section and into the database.
Alternative flow	o If the input isn't valid per the type requirements, the system issues a message for the student to enter the correct value.

7.4.2 Use Case Diagram

give name to the actor



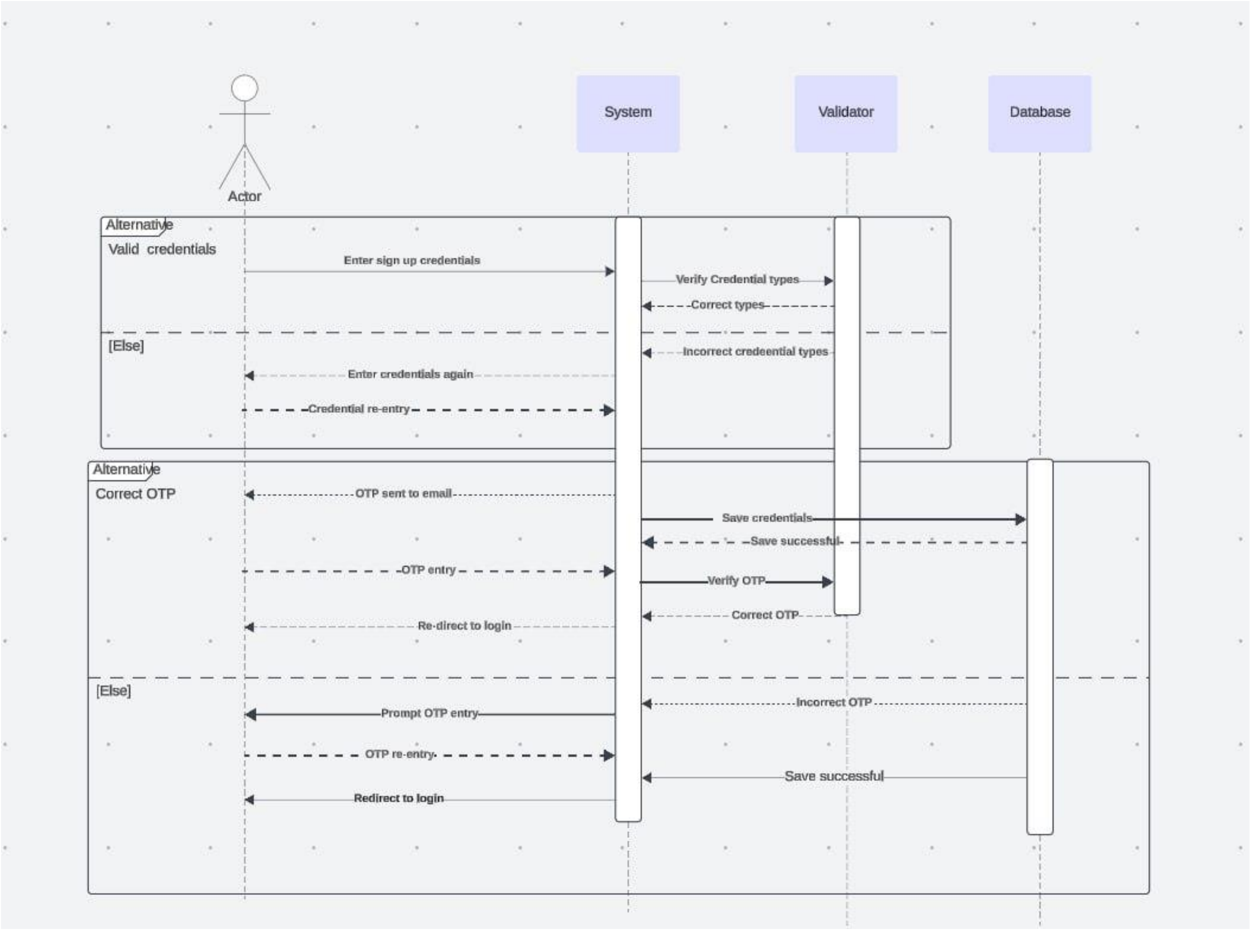
7.4.3 Dynamic Model

7.4.3.1 Sequence Diagram

7.4.3.1.1 UC001 - registerUser

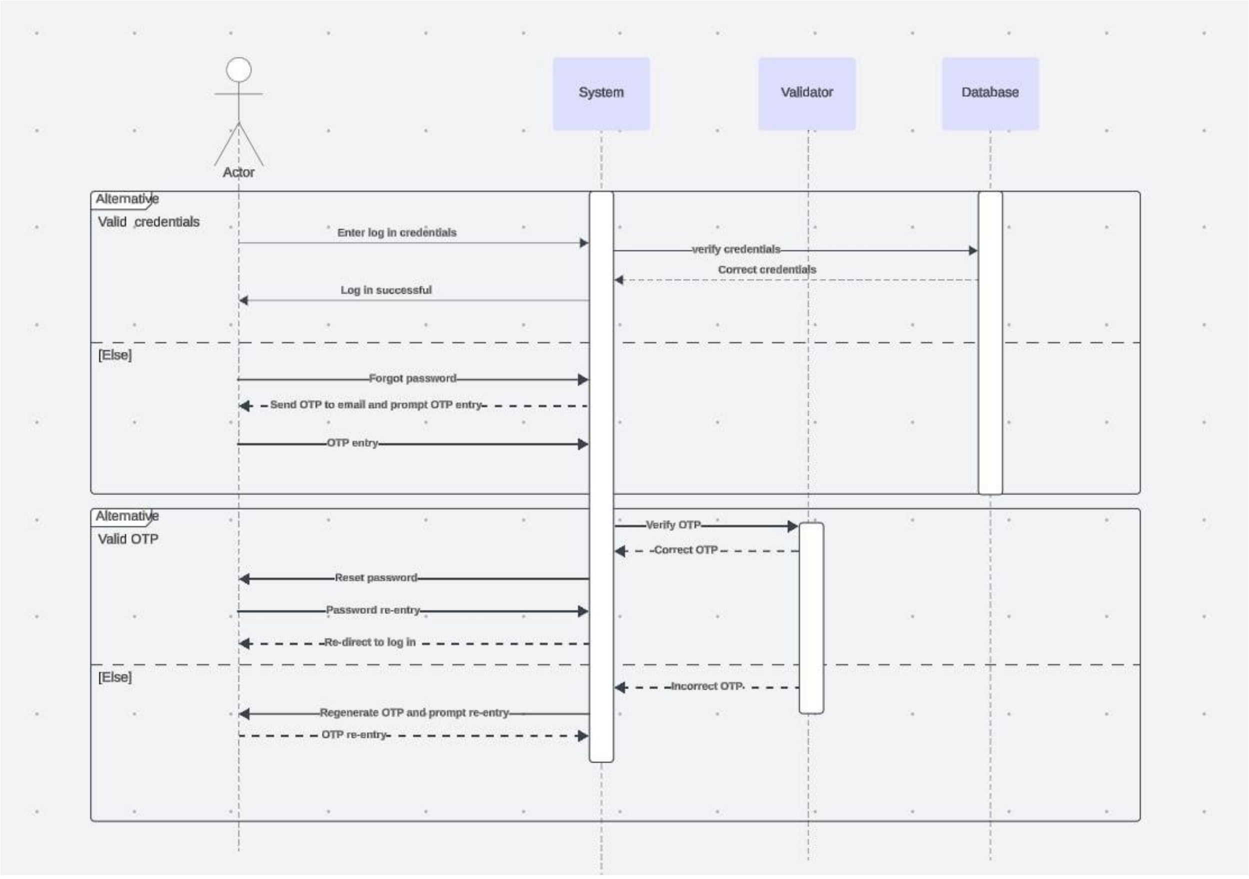
Start with identification of entity objects, boundary objects and control objects describe them in a table for every sequence diagram

Comment: The validator uses the validateInput use case.

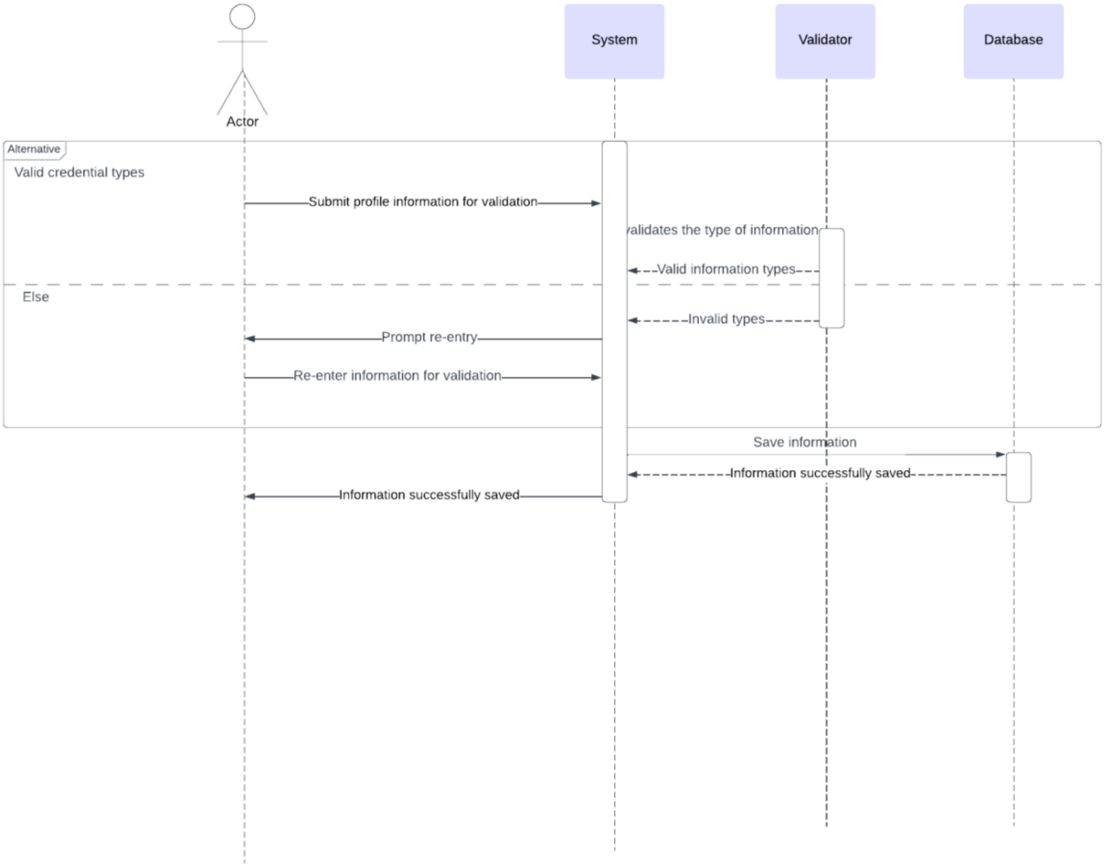


7.4.3.1.2 UC002 - authenticateUser

Comment: The validator uses the validateInput use case.



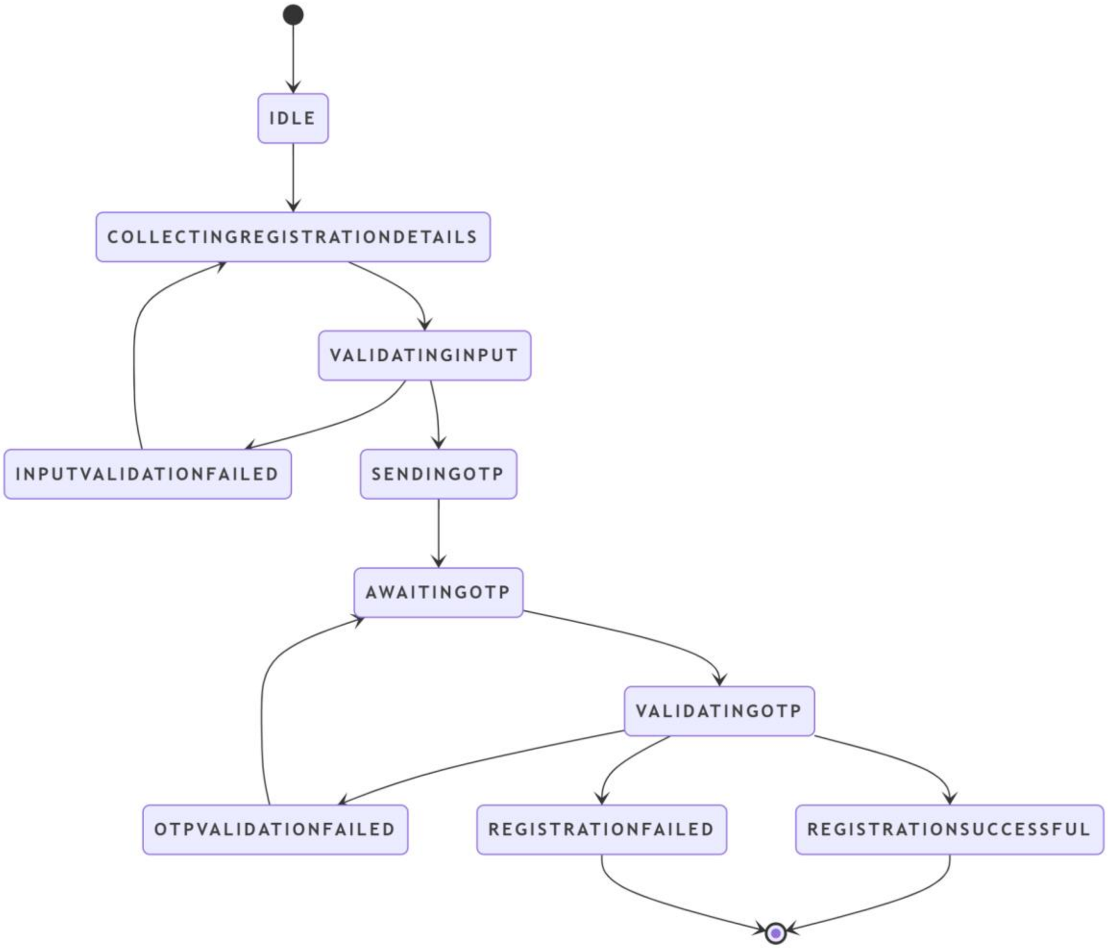
7.4.3.1.3 UC004 - editProfile


System
Validator
Database
Actor
Alternative
Valid credential types
Submit profile information for validation
validates the type of information
Valid information types
Else
Invalid types
Prompt re-entry
Re-enter information for validation
Save information
Information successfully saved
Information successfully saved

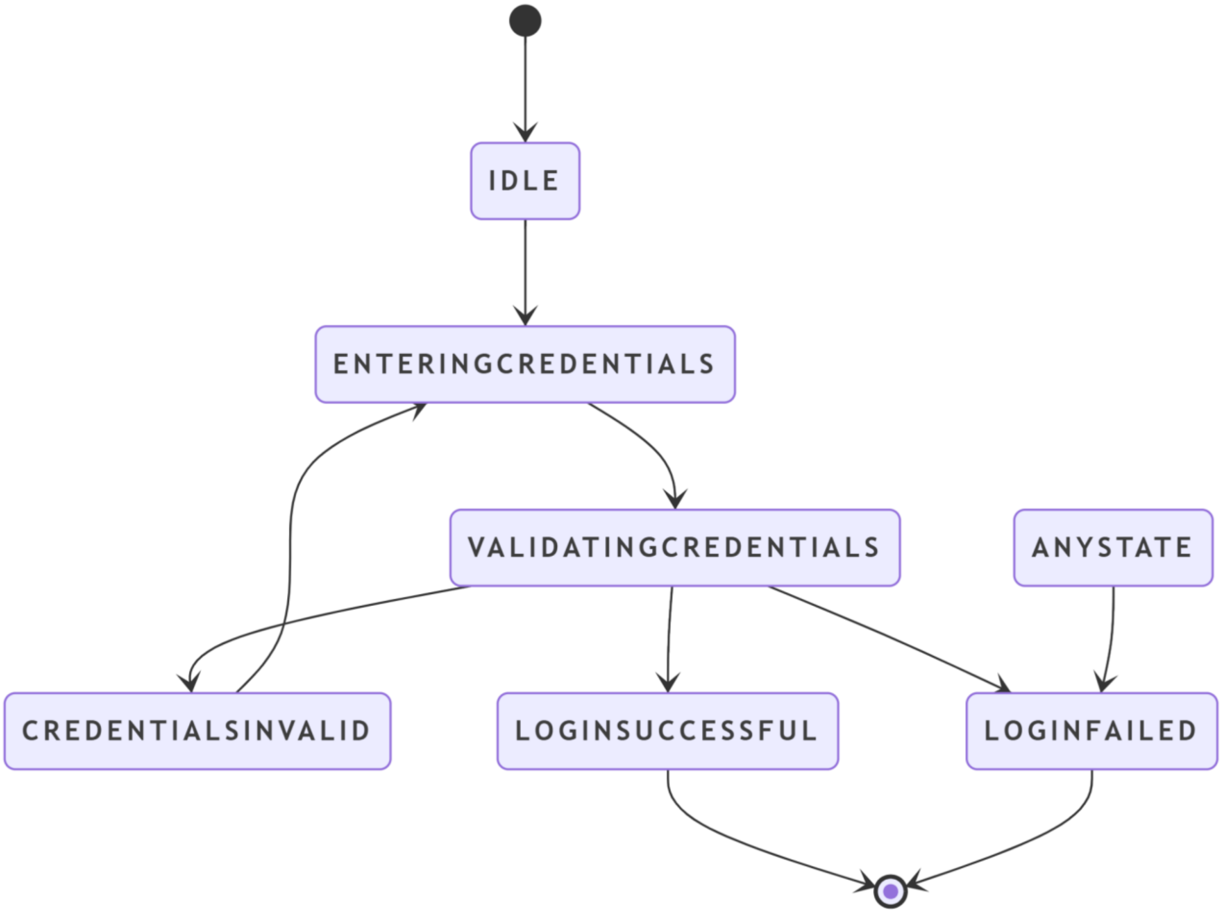
7.4.3.2 State Diagram

7.4.3.2.1 registerUser

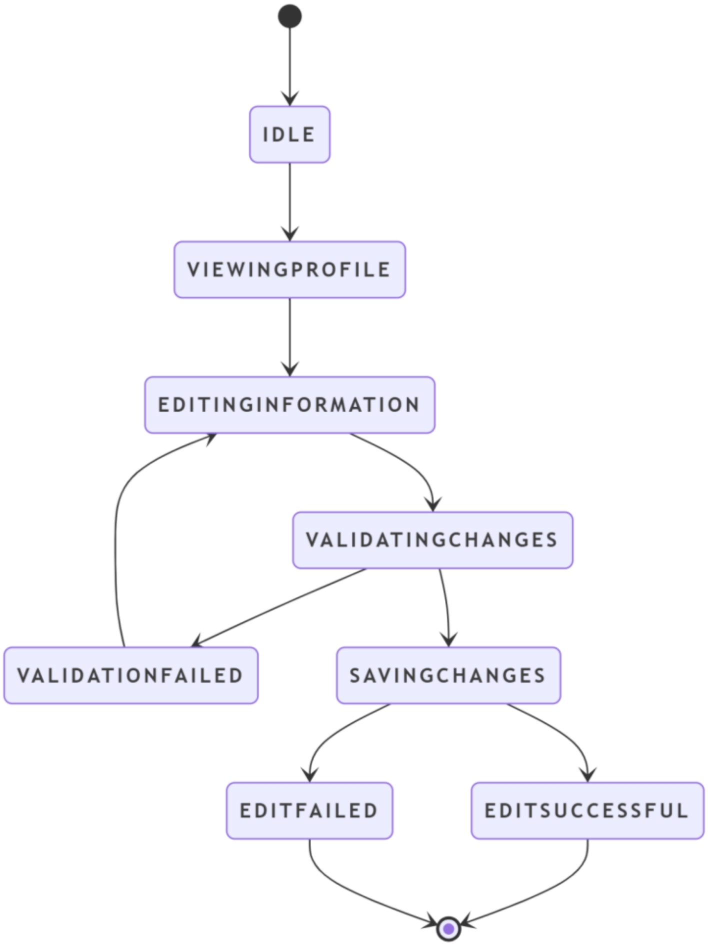
the terms represent action not state, use nouns for state



7.4.3.2.2 authenticateUser

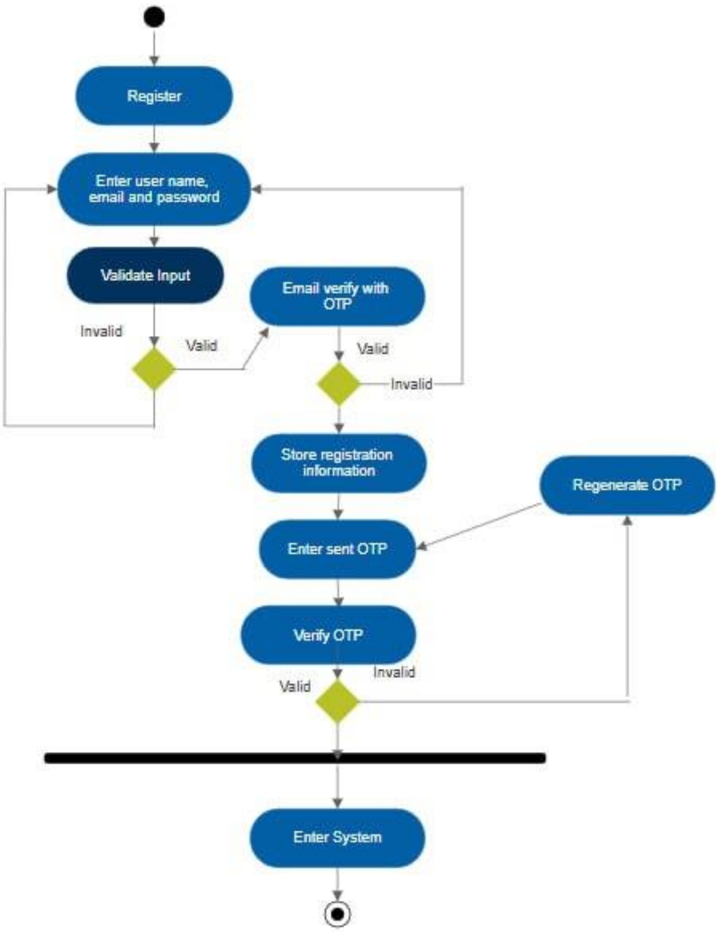


7.4.3.2.3 editInformation

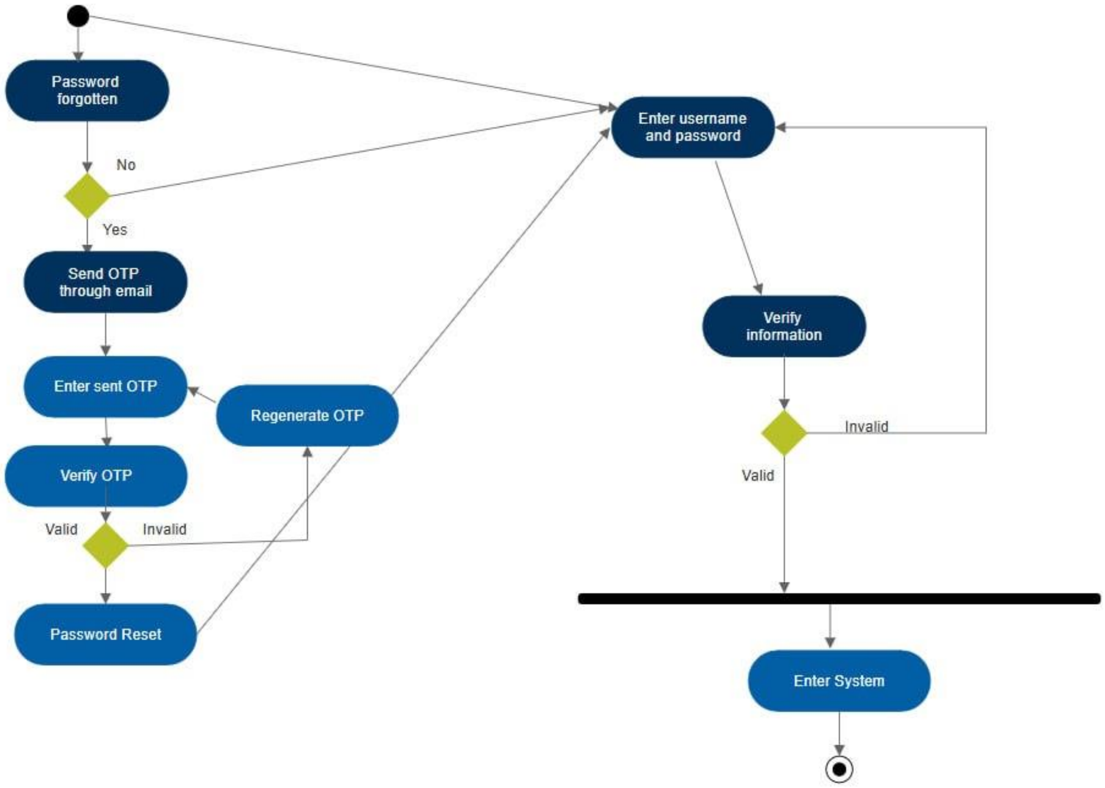


7.4.3.3 Activity Diagram

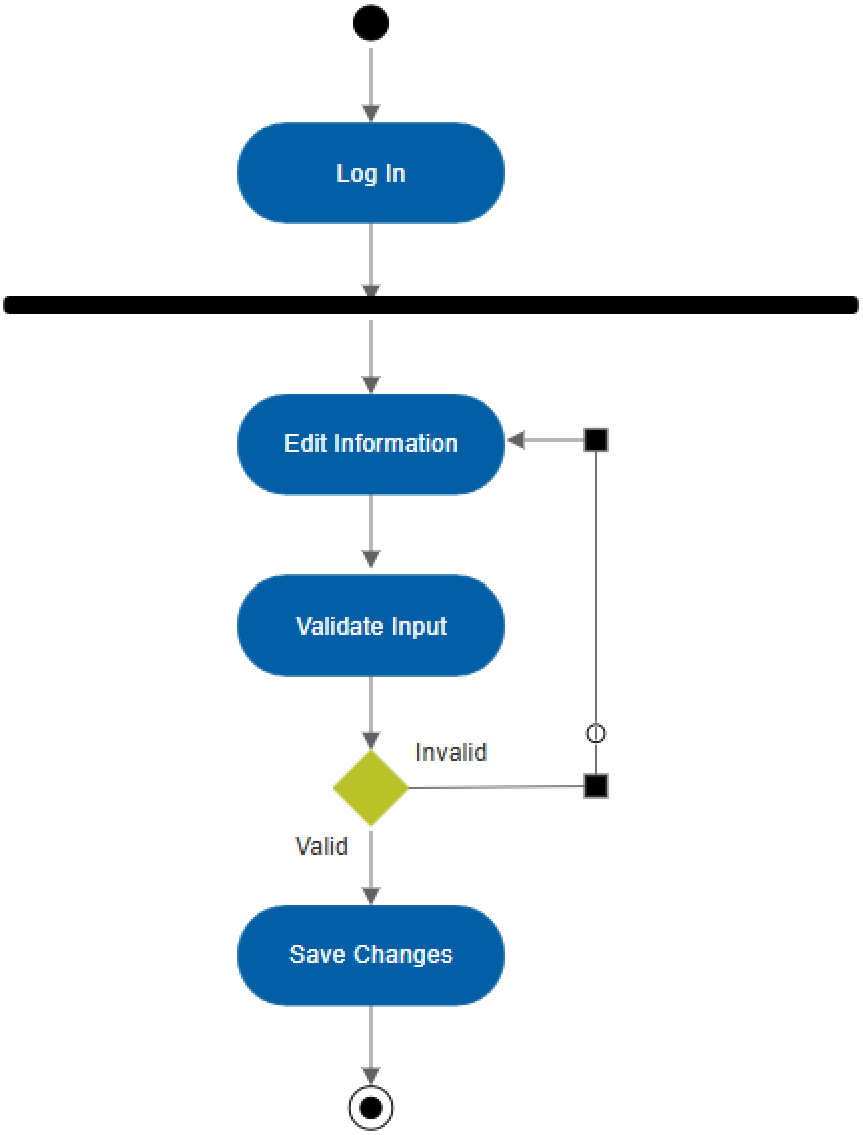
7.4.3.3.1 registerUser



7.4.3.3.2 authenticateUser



7.4.3.3.3 editInformation



7.4.3.4 Object model

Class	Fields	Method
User	○ Long id ○ String username ○ String email ○ String password ○ String fristName ○ String lastName ○ Int targetScore	○ getId() ○ setId() ○ getUsername() ○ setUsername() ○ getPassword() ○ setPassword() ○ setFirstName() ○ getFirstName() ○ setLastName() ○ getLastName() ○ setTargetScore() ○ getTargetScore()
UserService	○ UserRepository	○ registerUser() ○ loginUser() ○ verifyAccount() ○ verifyAccountPasswordReset() ○ regenerateOtp() ○ forgetPassword() ○ resetPassword() ○ studentInfo() ○ getUserByEmail() ○ editUserInfo()
UserRepository	No field	○ findByEmail() ○ findByUsername()
EmailService	○ JavaMailSender ○ String mailHost ○ String mailPort ○ String mailUsername ○ String mailPassword	○ sendOtpEmail() ○ getJavaMailSender()

7.4.3.5 Relationships

User ↔ UserService:

- UserService interacts with the user entity to perform operations like registration, login, and profile updates.

UserService ↔ UserRepository:

- UserService relies on userRepository to interact with the database.

UserRepository ↔ User:

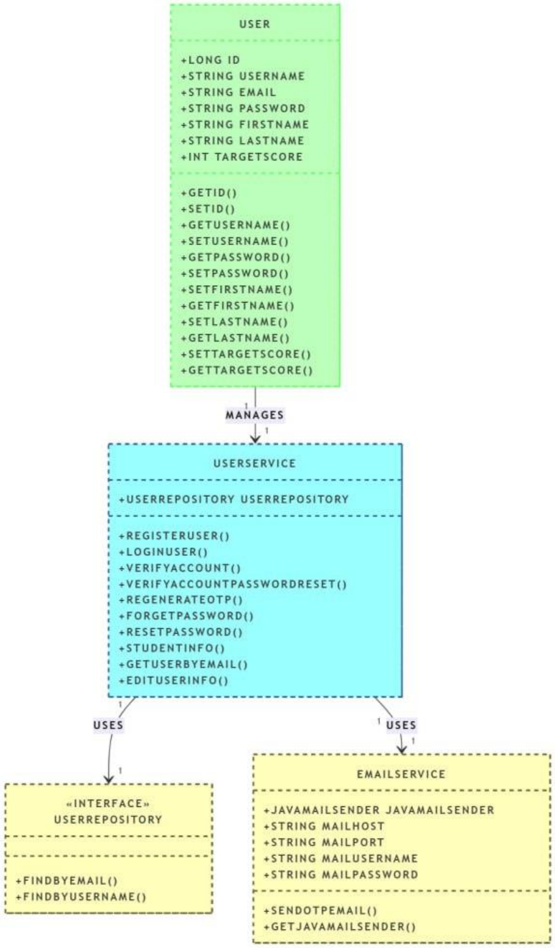
- UserRepository is the Data Access Object (DAO) for the user entity.

UserService ↔ EmailService:

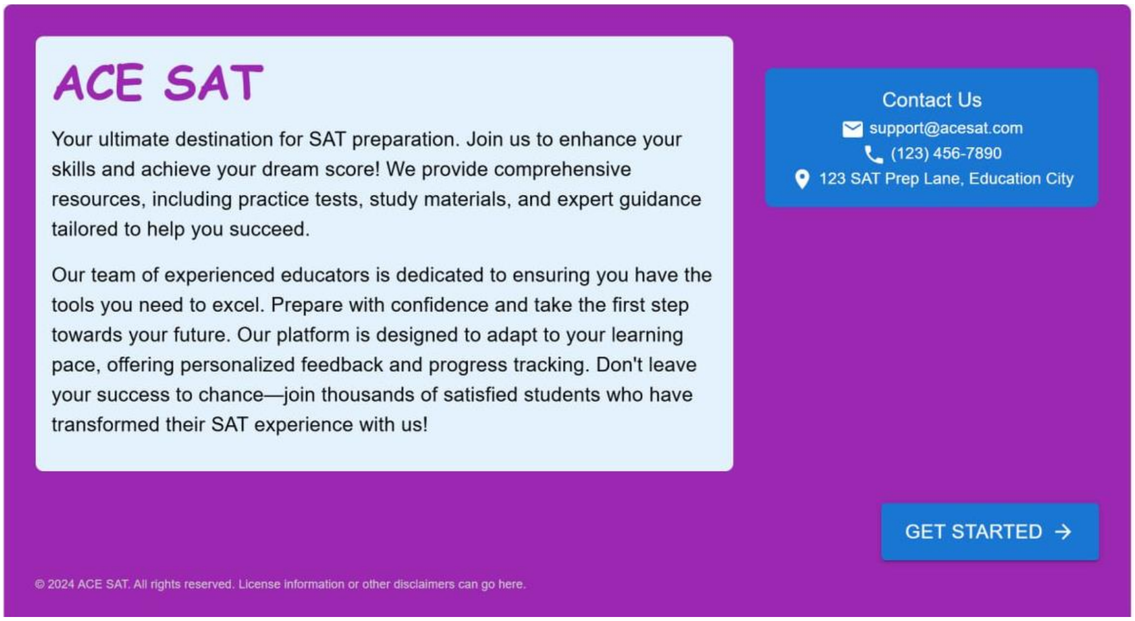
- UserService uses EmailService to handle user account verification, OTP emails, and password reset notifications.

EmailService ↔ External Mail Server:

- EmailService uses JavaMailSender to connect to an SMTP(Simple Mail Transfer Protocol) server.



7.4.3.6 User Interface



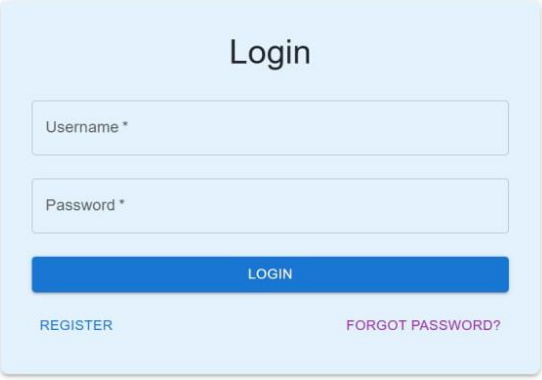
ACE SAT

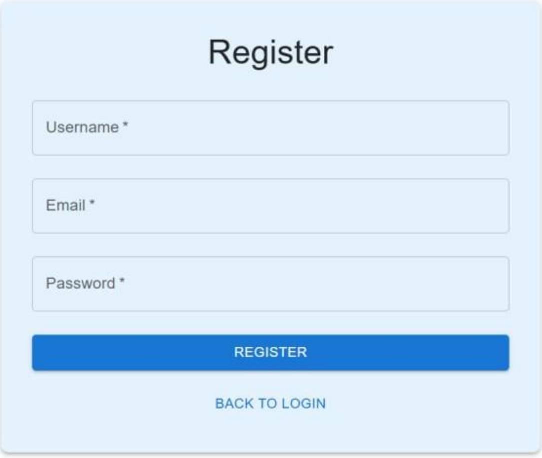
Your ultimate destination for SAT preparation. Join us to enhance your skills and achieve your dream score! We provide comprehensive resources, including practice tests, study materials, and expert guidance tailored to help you succeed.

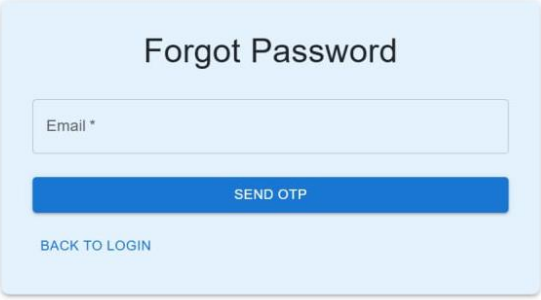
Our team of experienced educators is dedicated to ensuring you have the tools you need to excel. Prepare with confidence and take the first step towards your future. Our platform is designed to adapt to your learning pace, offering personalized feedback and progress tracking. Don't leave your success to chance—join thousands of satisfied students who have transformed their SAT experience with us!

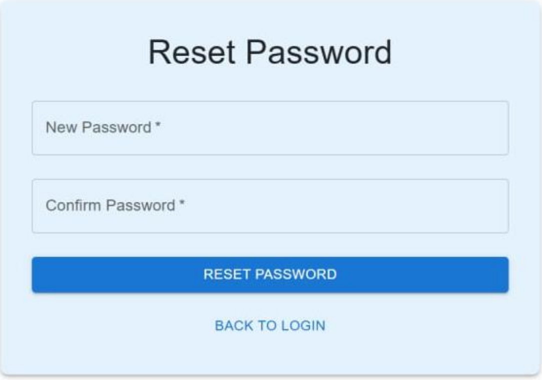
Contact Us
support@acesat.com
(123) 456-7890
123 SAT Prep Lane, Education City

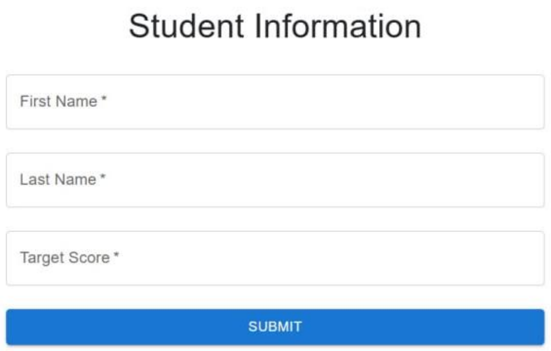
© 2024 ACE SAT. All rights reserved. License information or other disclaimers can go here.



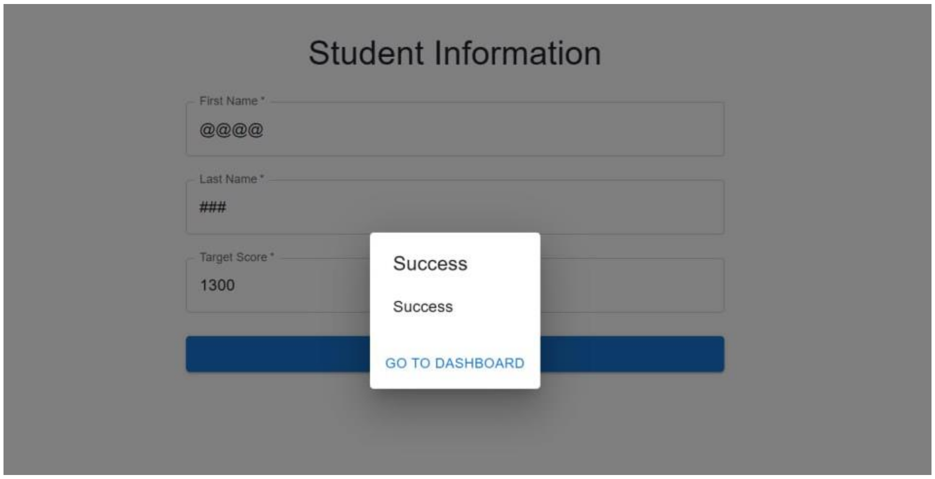








Student Information



Student Information

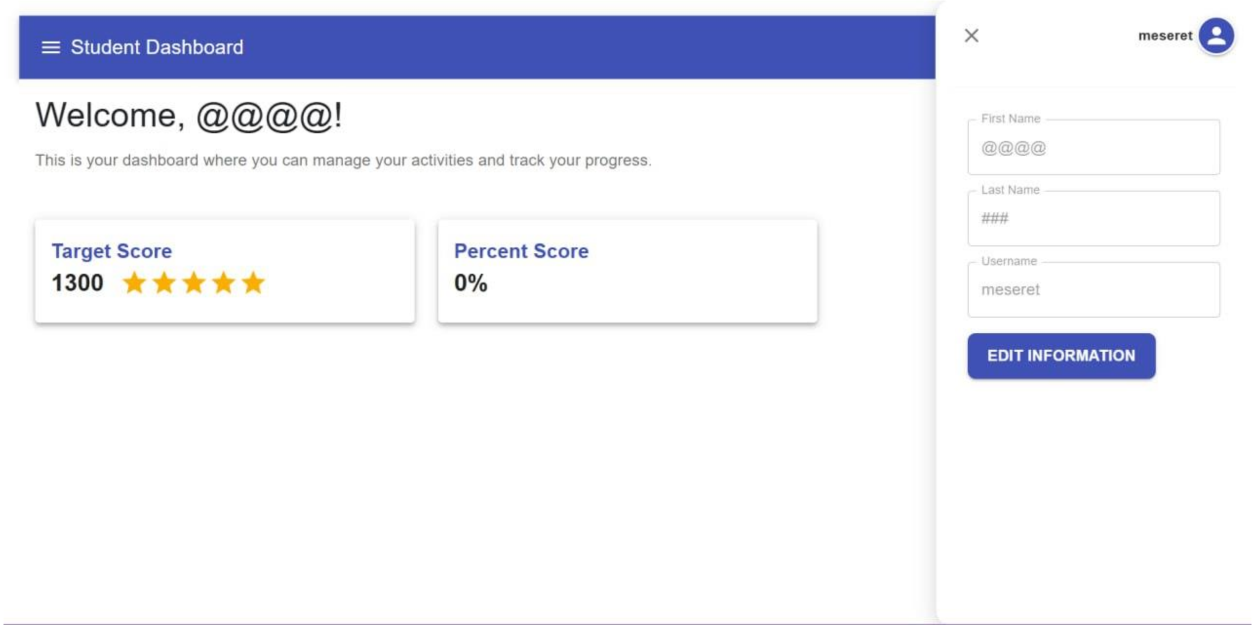
First Name * @@@@

Last Name * ###

Target Score * 1300

Success

Success



Welcome, @@@@!

This is your dashboard where you can manage your activities and track your progress.

Target Score 1300 ★★★★★

Percent Score 0%

First Name @@@@

Last Name ###

Username meseret