

Software Requirement Specification

for
SUBuddy

Prepared by team Amigos

-Sharvari Satish Doijode

-Anish Kuvelkar

-Siva Sai Priyatam Kilari

Syracuse University CSE 687

April 8,2024

Version	Date	A/D/C (Add, Delete Author , Change)	Document Section #	Description
1	April 8, 2024		Sharvari Doijode	

Table of Contents

1 INTRODUCTION.....	9
1.1 Purpose	9
1.2 Scope.....	9
1.3 Definition	9
1.4 Reference	9
1.5 Overview	9
2 Overall Description	10
2.1 Product Perspective	10
2.1.1 SUBuddy_BDD.....	10
2.1.2 SUBuddyInteraction.....	10
2.1.3 UserRegistration Microservice.....	10
2.1.4 UserAuthentication MicroService.....	10
2.1.5 Chat Microservice.....	10
2.1.6 Buddy/Alumini Search MicroService.....	11
2.2 Product Functions	11
2.2.1 User Login and Signup	11
2.2.1.1 Login and Sign Up.....	11
2.2.1.2 Student.....	11
2.2.1.2.1 Properties.....	12
2.2.1.2.2 Relationships.....	12
2.2.1.3 Enter Credentials.....	12
2.2.1.3.1 Properties.....	12
2.2.1.3.2 Relationships.....	12
2.2.1.4 Enter details and information.....	12
2.2.1.4.1 Properties.....	12
2.2.1.4.2 Relationships.....	12
2.2.1.5 Login.....	13
2.2.1.6 Signup.....	13
2.2.1.7 Validate User Details.....	13
2.2.1.8 Validate user credentials.....	13
2.2.1.9 Signup As Student/ Alumni Requirement Diagram.....	14
2.2.1.10 Signup or Login As Student/ Alumni.....	14
2.2.1.11 If Login.....	14
2.2.1.12 If Signup.....	15
2.2.1.13 Student should be SUstudent.....	15
2.2.1.14 Validate user.....	15
2.2.2 User Authentication.....	15
2.2.2.1 UserAuthentication	15
2.2.2.2 Student.....	16
2.2.2.2.1 Properties.....	16
2.2.2.2.2 Relationships.....	16
2.2.2.3 Enter credentials.....	16
2.2.2.3.1 Properties.....	16
2.2.2.4 OTP sent to the Email.....	16

2.2.2.4.1 Properties.....	16
2.2.2.4.2 Relationships.....	16
2.2.2.5 User Authentication.....	17
2.2.2.6 User validated.....	17
2.2.2.6.1 Properties.....	17
2.2.2.7 Manage Profile Requirement Diagram.....	17
2.2.2.8 Authentication and Manage Profile.....	17
2.2.2.9 Valid user account.....	17
2.2.2.10 Validate user.....	18
2.2.2.11 Chat service.....	18
2.2.2.12 Student.....	18
2.2.2.12.1 Properties.....	18
2.2.2.12.2 Relationships.....	18
2.2.2.13 Added to microsoft teams group.....	19
2.2.2.13.1 Properties.....	19
2.2.2.14 Chat with user.....	19
2.2.2.15 Join Groups.....	19
2.2.2.16 Login to MS teams.....	20
2.2.2.16.1 Properties.....	20
2.2.2.17 Select user to chat with.....	20
2.2.2.17.1 Properties.....	20
2.2.2.18 Chat with another Student/ Alumni Requirement Diagram.....	21
2.2.2.19 Chat with another Student/ Alumni.....	21
2.2.2.20 User account present.....	21
2.2.2.21 Messages.....	21
2.2.2.22 Valid Student search.....	21
2.2.2.23 Buddy /Alumni search.....	22
2.2.2.24 Refine data of students user wants to view.....	22
2.2.2.24.1 Properties.....	22
2.2.2.25 Buddy Search.....	22
2.2.2.26 Login.....	23
2.2.2.27 Student.....	23
2.2.2.27.1 Properties.....	23
2.2.2.27.2 Relationships.....	23
2.2.2.28 Student.....	23
2.2.2.28.1 Properties.....	23
2.2.2.28.2 Relationships.....	23
2.2.2.29 Student.....	24
2.2.2.29.1 Properties.....	24
2.2.2.29.2 Relationships.....	24
2.2.2.30 User fills questionare.....	24
2.2.2.30.1 Properties.....	24
2.2.2.30.2 Relationships.....	24
2.2.2.31 User redirected to Buddy Search page.....	24
2.2.2.31.1 Properties.....	24
2.2.2.31.2 Relationships.....	25
2.2.2.32 User redirected to home page.....	25
2.2.2.32.1 Properties.....	25
2.2.2.32.2 Relationships.....	25
2.2.2.33 Join / Leave groups Requirement Diagram.....	25
2.2.2.34 Search buddies Join / Leave groups.....	25

2.2.2.35 Account present.....	26
2.2.2.36 Filter Group.....	26
2.2.2.37 Participant of group.....	26
3 Specific Requirements	26
3.1 Overview.....	26
3.1.1 Signup Activity Diagram.....	26
3.1.2 Signup Activity	27
3.1.2.1 Sub Diagrams.....	27
3.1.2.1.1 Signup System Requirement Diagram.....	27
3.1.3 unnamed.....	27
3.1.3.1 Properties.....	27
3.1.3.2 Relationships.....	27
3.1.4 Enter User Information.....	27
3.1.4.1 Properties.....	28
3.1.4.2 Relationships.....	28
3.1.5 Answer the pre-defined questions.....	28
3.1.5.1 Properties.....	28
3.1.5.2 Relationships.....	28
3.1.6 Display Profile.....	28
3.1.6.1 Properties.....	28
3.1.6.2 Relationships.....	28
3.1.7 Start.....	29
3.1.7.1 Properties.....	29
3.1.7.2 Relationships.....	29
3.1.8 Store in Database.....	29
3.1.8.1 Properties.....	29
3.1.8.2 Relationships.....	29
3.1.9 Store in Database.....	29
3.1.9.1 Properties.....	29
3.1.9.2 Relationships.....	29
3.1.10 Valid Data?.....	30
3.1.10.1 Properties.....	30
3.1.10.2 Relationships.....	30
3.1.11 Validate Data.....	30
3.1.11.1 Properties.....	30
3.1.11.2 Relationships.....	30
3.1.12 login Activity Diagram.....	31
3.1.13 Login.....	31
3.1.13.1 Sub Diagrams.....	31
3.1.13.1.1 Login System Requirement Diagram.....	32
3.1.14 End.....	32
3.1.14.1 Properties.....	32
3.1.14.2 Relationships.....	32
3.1.15 Input email/ password.....	32
3.1.15.1 Properties.....	32
3.1.15.2 Relationships.....	32
3.1.16 Jump to Profile Page.....	33
3.1.16.1 Properties.....	33

3.1.16.2 Relationships.....	33
3.1.17 Sign up.....	33
3.1.17.1 Properties.....	33
3.1.17.2 Relationships.....	33
3.1.18 Start.....	33
3.1.18.1 Properties.....	33
3.1.18.2 Relationships.....	33
3.1.19 Validate Email with OTP.....	34
3.1.19.1 Properties.....	34
3.1.19.2 Relationships.....	34
3.1.20 email exist ?.....	34
3.1.20.1 Properties.....	34
3.1.20.2 Relationships.....	34
3.1.21 User Authentication.....	35
3.1.22 Authentication.....	35
3.1.22.1 Sub Diagrams.....	35
3.1.22.1.1 Authentication Requirement Diagram.....	36
3.1.23 unnamed.....	36
3.1.23.1 Properties.....	36
3.1.23.2 Relationships.....	36
3.1.24 unnamed.....	36
3.1.24.1 Properties.....	36
3.1.24.2 Relationships.....	36
3.1.25 Check if the emailid exists in Database.....	37
3.1.25.1 Properties.....	37
3.1.25.2 Relationships.....	37
3.1.26 Login.....	37
3.1.26.1 Properties.....	37
3.1.26.2 Relationships.....	37
3.1.27 OTP sent to student.....	37
3.1.27.1 Properties.....	37
3.1.27.2 Relationships.....	38
3.1.28 Redirect to signup page.....	38
3.1.28.1 Properties.....	38
3.1.28.2 Relationships.....	38
3.1.29 Student Enters OTP.....	38
3.1.29.1 Properties.....	38
3.1.29.2 Relationships.....	38
3.1.30 User Authenticated	39
3.1.30.1 Properties.....	39
3.1.30.2 Relationships.....	39
3.1.31 Users profile displayed.....	39
3.1.31.1 Properties.....	39
3.1.31.2 Relationships.....	39
3.1.32 Validate OTP.....	39
3.1.32.1 Properties.....	39
3.1.32.2 Relationships.....	39
3.1.33 Buddy/Alumni search.....	40

3.1.34 Buddy Alumni Search.....	40
3.1.34.1 Sub Diagrams.....	40
3.1.34.1.1 Search System Required Diagram.....	41
3.1.35 unnamed.....	41
3.1.35.1 Properties.....	41
3.1.35.2 Relationships.....	41
3.1.36 unnamed.....	41
3.1.36.1 Properties.....	41
3.1.36.2 Relationships.....	41
3.1.37 Data filtering processing.....	42
3.1.37.1 Properties.....	42
3.1.37.2 Relationships.....	42
3.1.38 Filtered students.....	42
3.1.38.1 Properties.....	42
3.1.38.2 Relationships.....	42
3.1.39 Login in to the account.....	42
3.1.39.1 Properties.....	42
3.1.39.2 Relationships.....	43
3.1.40 User goes to student/alumni search page.....	43
3.1.40.1 Properties.....	43
3.1.40.2 Relationships.....	43
3.1.41 user fills out questionnaire.....	43
3.1.41.1 Properties.....	43
3.1.41.2 Relationships.....	44
3.1.42 Chat services.....	44
3.1.43 unnamed.....	45
3.1.43.1 Properties.....	45
3.1.43.2 Relationships.....	45
3.1.44 Add Criteria and filters to join group.....	45
3.1.44.1 Properties.....	45
3.1.44.2 Relationships.....	45
3.1.45 Add Filters.....	45
3.1.45.1 Properties.....	46
3.1.45.2 Relationships.....	46
3.1.46 Chat.....	46
3.1.46.1 Properties.....	46
3.1.46.2 Relationships.....	46
3.1.47 Click "join group".....	46
3.1.47.1 Properties.....	46
3.1.47.2 Relationships.....	46
3.1.48 Close the chat.....	47
3.1.48.1 Properties.....	47
3.1.48.2 Relationships.....	47
3.1.49 End.....	47
3.1.49.1 Properties.....	47
3.1.49.2 Relationships.....	47
3.1.50 Login.....	47
3.1.50.1 Properties.....	47

3.1.50.2 Relationships.....	48
3.1.51 Peer to peer chat.....	48
3.1.51.1 Properties.....	48
3.1.51.2 Relationships.....	48
3.1.51.3 Sub Diagrams.....	48
3.1.51.3.1 Peer to peer chat System Requirement Diagram.....	49
3.1.52 Redirected to microsoft teams,new group chat created.....	49
3.1.52.1 Properties.....	49
3.1.52.2 Relationships.....	49
3.1.53 Start.....	49
3.1.53.1 Properties.....	50
3.1.53.2 Relationships.....	50
3.2 Performance Requirement.....	50
3.2.1 Login and Signup.....	50
3.2.2 Compatibility.....	50
3.2.3 Password Recovery.....	50
3.2.4 Secure Authentication.....	50
3.2.5 Session Management.....	50
3.2.6 User-Friendly Interface.....	50
3.2.7 Login and Signup.....	50
3.2.8 Account Confirmation.....	50
3.2.9 Duplicate Account Check.....	51
3.2.10 Email Verification.....	51
3.2.11 Password Strength Criteria.....	51
3.2.12 User Registration Form.....	51
3.2.13 UserAuthentication MicroService.....	51
3.2.14 User-Friendly Error Messages.....	51
3.2.15 Authentication Interface.....	51
3.2.16 Authentication Success Confirmation.....	51
3.2.17 Credential Validation.....	51
3.2.18 Two-Factor Authentication.....	52
3.2.19 Buddy/Alumini Search MicroService.....	52
3.2.20 Error Handling in Search.....	52
3.2.21 Filter Search.....	52
3.2.22 Search History.....	52
3.2.23 Search Results Display.....	52
3.2.24 Student Search.....	52
3.2.25 Chat Microservice.....	52
3.2.26 File Sharing in Chat.....	52
3.2.27 Instant Messaging.....	53
3.2.28 Message Storage.....	53
3.2.29 Notification System.....	53
3.2.30 SUBuddyInteraction.....	53

3.2.31 Buddy/Alumni Search Microservice:.....	53
3.2.32 Chat Microservice:.....	53
3.2.33 UserAuthentication Microservice.....	53
3.2.34 UserRegistration Microservice.....	53
3.3 Design Constraints.....	54
3.3.1 SUBuddyInteraction.....	54
3.3.2 Buddy/Alumni Search Microservice:.....	54
3.3.3 Chat Microservice:.....	54
3.3.4 UserAuthentication Microservice:.....	54
3.3.5 UserRegistration Microservice:.....	54

1. INTRODUCTION

1.1. Purpose

The purpose of the SUBuddy application is to provide an interactive platform dedicated to enhancing the social and academic lives of Syracuse University students. It aims to bridge the gap between international students and the wider university community by fostering social connections, encouraging cultural exchange, and facilitating academic support and networking opportunities.

1.2. Scope

SUBuddy is scoped to include a comprehensive suite of features:

1. User Authentication: Secure sign-up and login processes.
2. Buddy System: An algorithm-based matching system to suggest potential buddies.
3. Communication: Real-time chat functionalities to connect students.
4. Profile Management: A personal space for students to manage their profiles.
5. Community Engagement: Forums for discussions, group activities, and event management.

User Experience: Intuitive interface design, accessibility compliance, and customizable settings.

Support Infrastructure: A robust help and support system for user assistance

1.3. Definition

BDD - Block Definition Diagram

SRS- Software Specification requirements

1.4. Reference

IEEE Std 830-1998 - IEEE Recommended Practice for Software Requirements Specifications - Revision of IEEE Std 830-1993

The development of SUBuddy will adhere to the following references and standards:

1. WCAG 2.1: Ensuring web content is accessible and inclusive.
2. OOP Principles: The four pillars of Object-Oriented Programming—Encapsulation, Inheritance, Polymorphism, and Abstraction—that inform the application architecture.
3. Firebase: The backend infrastructure for hosting and data management.
4. Web Development Tools: The technologies used for crafting the front-end experience.

1.5. Overview

SUBuddy stands out as an innovative social networking application that is finely tuned to the needs of the Syracuse University student community. The application's features are carefully designed to address the unique challenges faced by students, particularly those studying away from their home countries. This document outlines the project proposal, system requirements, and the way forward for the development, deployment, and iteration of the SUBuddy application.

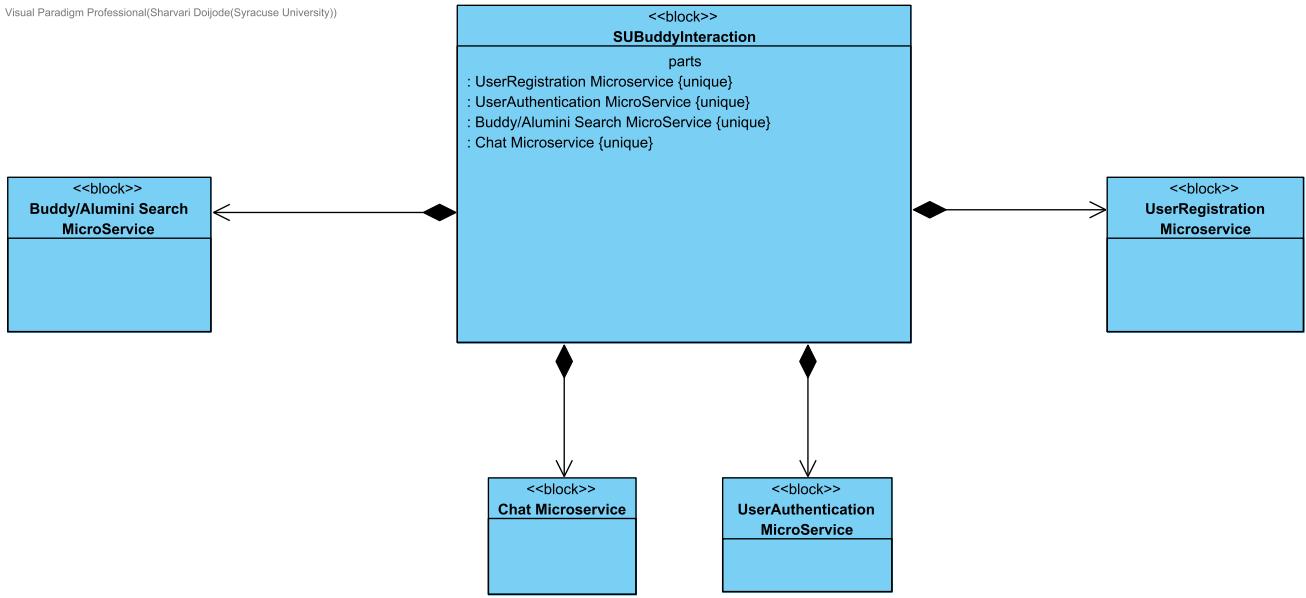
2. Overall Description

2.1. Product Perspective

SUBuddy is a social networking platform designed specifically for international students studying at Syracuse University. It aims to foster a supportive and inclusive community by providing tools and features that facilitate connections, collaborations, and cultural exchange among students from diverse backgrounds. Figure 1.1 Shows the Block definition diagram for SUBuddy. The microservices to be implemented are mentioned in this diagram.

2.1.1. SUBuddy_BDD

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



2.1.2. SUBuddyInteraction

Central system block that orchestrates the interaction between various microservices involved in the SUBuddy platform, including user registration, authentication, search, and chat functionalities.

Responsibilities:

1. Coordinate microservices interaction.
2. Ensure data consistency across services.
3. Provide a unified interface for the front-end to interact with backend services.

2.1.3. UserRegistration Microservice

Handles the creation of new user accounts, ensuring that all registration details are valid and securely stored. Sends confirmation emails post-registration.

2.1.4. UserAuthentication MicroService

Authenticates user credentials against the database records, manages active sessions, and issues authentication tokens for session management.

2.1.5. Chat Microservice

Manages real-time messaging between users, storing and retrieving message histories, and supports notifications for new messages

2.1.6. Buddy/Alumini Search MicroService

Enables users to find and connect with other alumni or buddies using search criteria. Integrates with the user profile database to retrieve and filter search results.

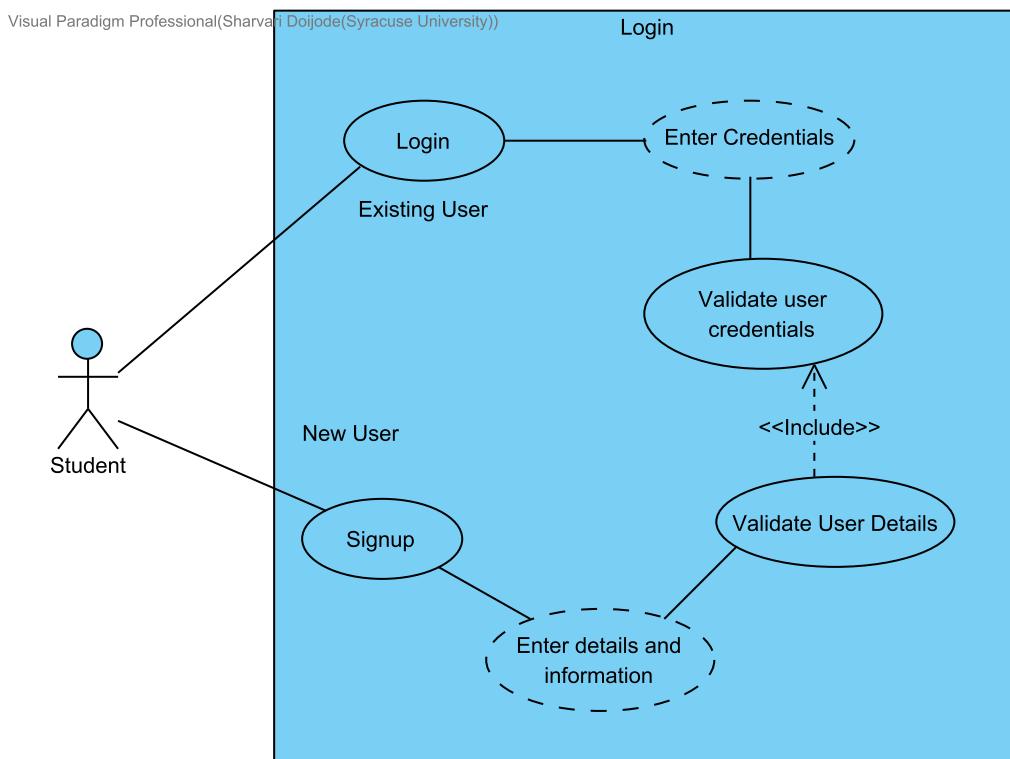
2.2. Product Functions

Below are the use case diagrams for the use cases of SUBuddy and the ways in which the user would be interacting with the system.

2.2.1. User Login and Signup

This use case specifies the user registration and login scenario. The use case specifies how user reacts with the system and what all elements would be included.

2.2.1.1. Login and Sign Up



This use case describes the process by which a potential user can register for a new account in the system. It involves inputting personal information, validating that information, creating the user account, and confirming the account through an email verification process. The system ensures that all user information is kept secure and that the user is informed of the progress throughout the registration process. If the user exists already the user will be prompted to login via email and password.

Figure 2.1 Login/ Register for user to create account in SUbuddy

2.2.1.2. Student

ID: UC02.AC10

2.2.1.2.1. Properties

Abstract	false
Leaf	false
Root	false

2.2.1.2.2. Relationships

Relationship	From	To
— Existing User	 Student	 Login
— New User	 Student	 Signup
— unnamed	 Student	 Join Groups
— unnamed	 Student	 User Authentication
— unnamed	 Student	 Chat with user

2.2.1.3. Enter Credentials

2.2.1.3.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.1.3.2. Relationships

Relationship	From	To
— unnamed	 Enter Credentials	 Validate user credentials
— unnamed	 Login	 Enter Credentials

2.2.1.4. Enter details and information

2.2.1.4.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.1.4.2. Relationships

Relationship	From	To
— unnamed	 Enter details and information	 Validate User Details
— unnamed	 Signup	 Enter details and information

2.2.1.5. Login

ID: UC19

Primary Actors	 Student
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

2.2.1.6. Signup

ID: UC20

Primary Actors	 Student
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

2.2.1.7. Validate User Details

ID: UC24

Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

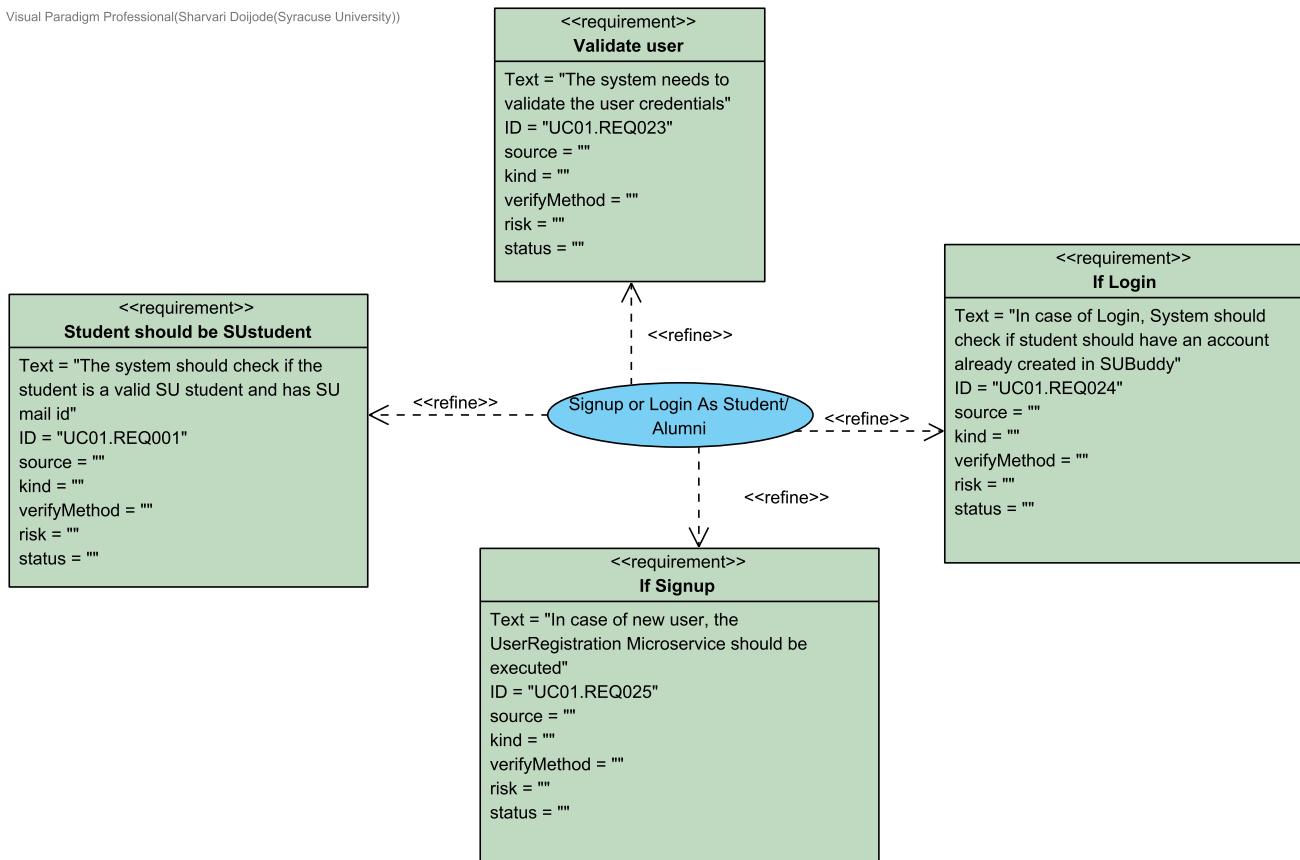
2.2.1.8. Validate user credentials

ID: UC21

Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

2.2.1.9. Signup As Student/ Alumni Requirement Diagram

Visual Paradigm Professional(Sharvari Dojode(Syracuse University))



2.2.1.10. Signup or Login As Student/ Alumni

ID: UC01

This use case describes the process by which a potential user can register for a new account in the system. It involves inputting personal information, validating that information, creating the user account, and confirming the account through an email verification process. The system ensures that all user information is kept secure and that the user is informed of the progress throughout the registration process.

2.2.1.11. If Login

ID: UC01.REQ024

In case of Login, System should check if student should have an account already created in SUBuddy

2.2.1.12. If Signup

ID: UC01.REQ025

In case of new user, the UserRegistration Microservice should be executed

2.2.1.13. Student should be SUstudent

ID: UC01.REQ001

The system should check if the student is a valid SU student and has SU mail id

2.2.1.14. Validate user

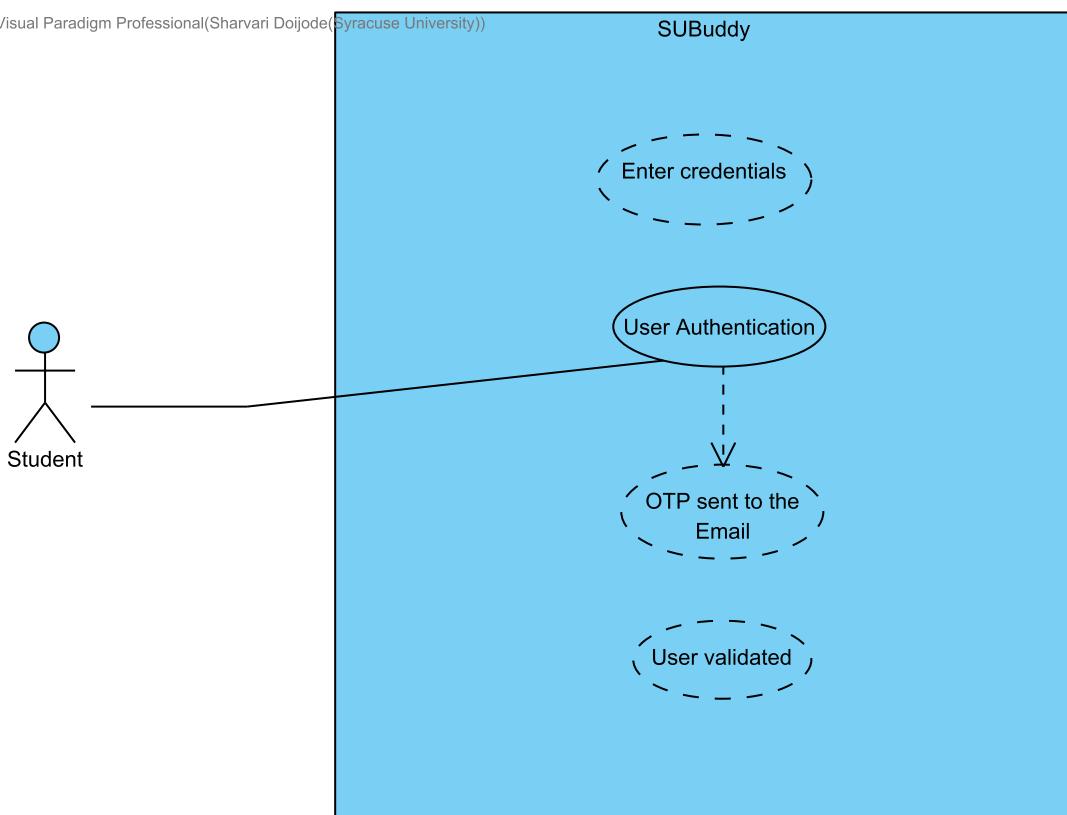
ID: UC01.REQ023

The system needs to validate the user credentials

2.2.2. User Authentication

2.2.2.1. UserAuthentication

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



This use case encompasses the steps required for a user to authenticate themselves to gain access to the system. The user must provide credentials, which the system validates against stored records. Upon successful authentication, the system issues a token or session identifier, which is then used for maintaining the user's session and securing further interactions within the system

2.2.2.2. Student

ID: UC02.AC10

2.2.2.2.1. Properties

Abstract	false
Leaf	false
Root	false

2.2.2.2.2. Relationships

Relationship	From	To
— Existing User	 Student	 Login
— New User	 Student	 Signup
— unnamed	 Student	 Join Groups
— unnamed	 Student	 User Authentication
— unnamed	 Student	 Chat with user

2.2.2.3. Enter credentials

2.2.2.3.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.4. OTP sent to the Email

2.2.2.4.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.4.2. Relationships

Relationship	From	To
→>unnamed	 User Authentication	 OTP sent to the Email

● 2.2.2.5. User Authentication

ID: UC35

Primary Actors	 Student
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

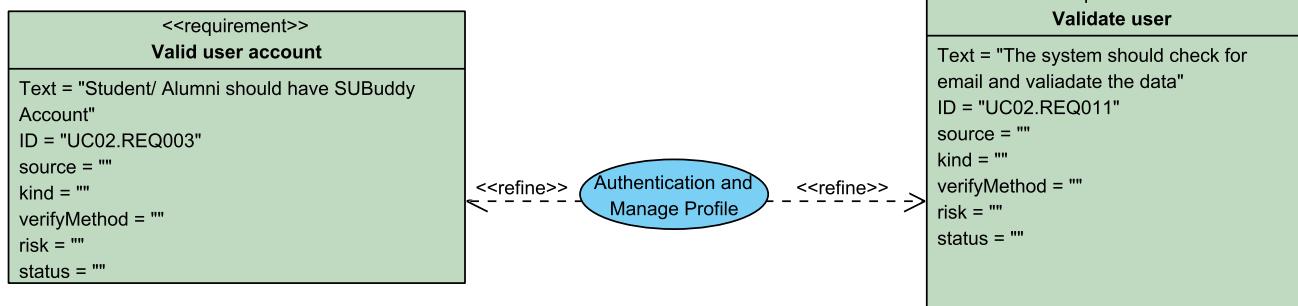
● 2.2.2.6. User validated

2.2.2.6.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.7. Manage Profile Requirement Diagram

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



● 2.2.2.8. Authentication and Manage Profile

ID: UC02

This use case describes the process by which a user identifies themselves to the SUBuddy system to gain access to their account and associated features. It includes entering credentials, validating them against the system, and handling successful or failed authentication attempts

2.2.2.9. Valid user account

ID: UC02.REQ003

Student/ Alumni should have SUBuddy Account

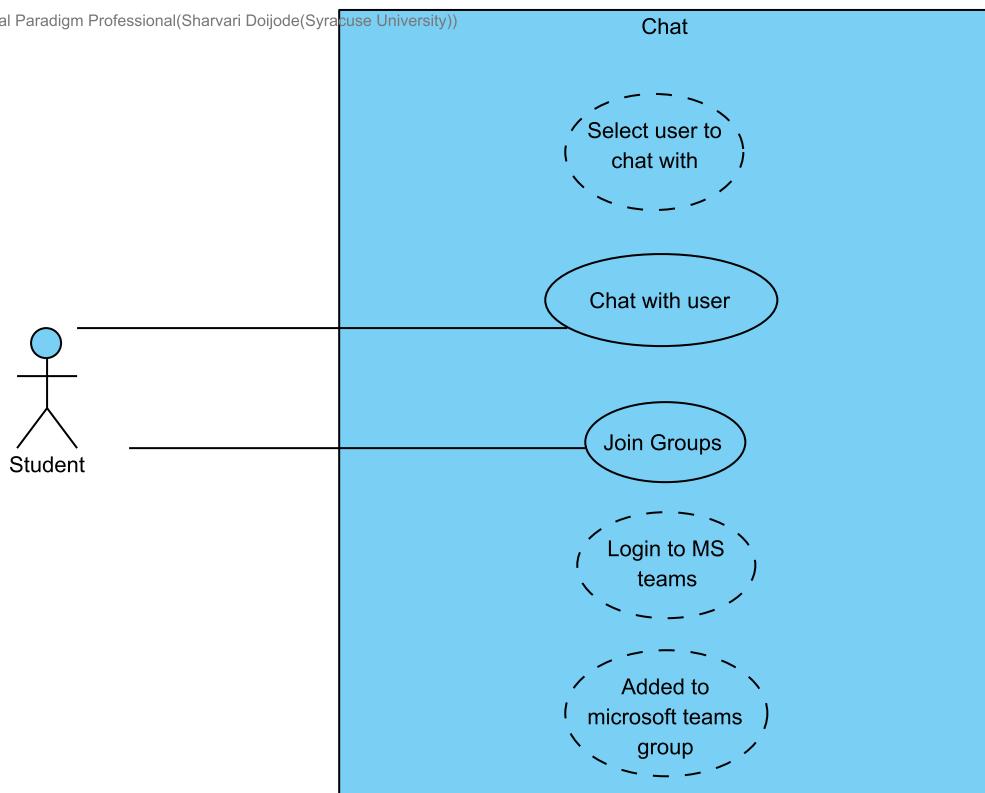
2.2.2.10. Validate user

ID: UC02.REQ011

The system should check for email and validate the data

2.2.2.11. Chat service

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



This use case outlines the ability for users to communicate with one another through a real-time chat interface. It covers the initiation of chat sessions, message composition, and the transmission of messages between users. The system is responsible for the delivery of messages and ensuring that users are notified of new messages, fostering immediate and continuous communication.

2.2.2.12. Student

ID: UC02.AC10

2.2.2.12.1. Properties

Abstract	false
Leaf	false
Root	false

2.2.2.12.2. Relationships

Relationship	From	To
— Existing User	>User	Login
— New User	User	Signup
— unnamed	User	Join Groups
— unnamed	User	User Authentication
— unnamed	User	Chat with user

● 2.2.2.13. Added to microsoft teams group

2.2.2.13.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

● 2.2.2.14. Chat with user

ID: UC26

Primary Actors	User
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

● 2.2.2.15. Join Groups

ID: UC27

Primary Actors	User
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A

Assumptions	N/A
--------------------	-----

● 2.2.2.16. *Login to MS teams*

2.2.2.16.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

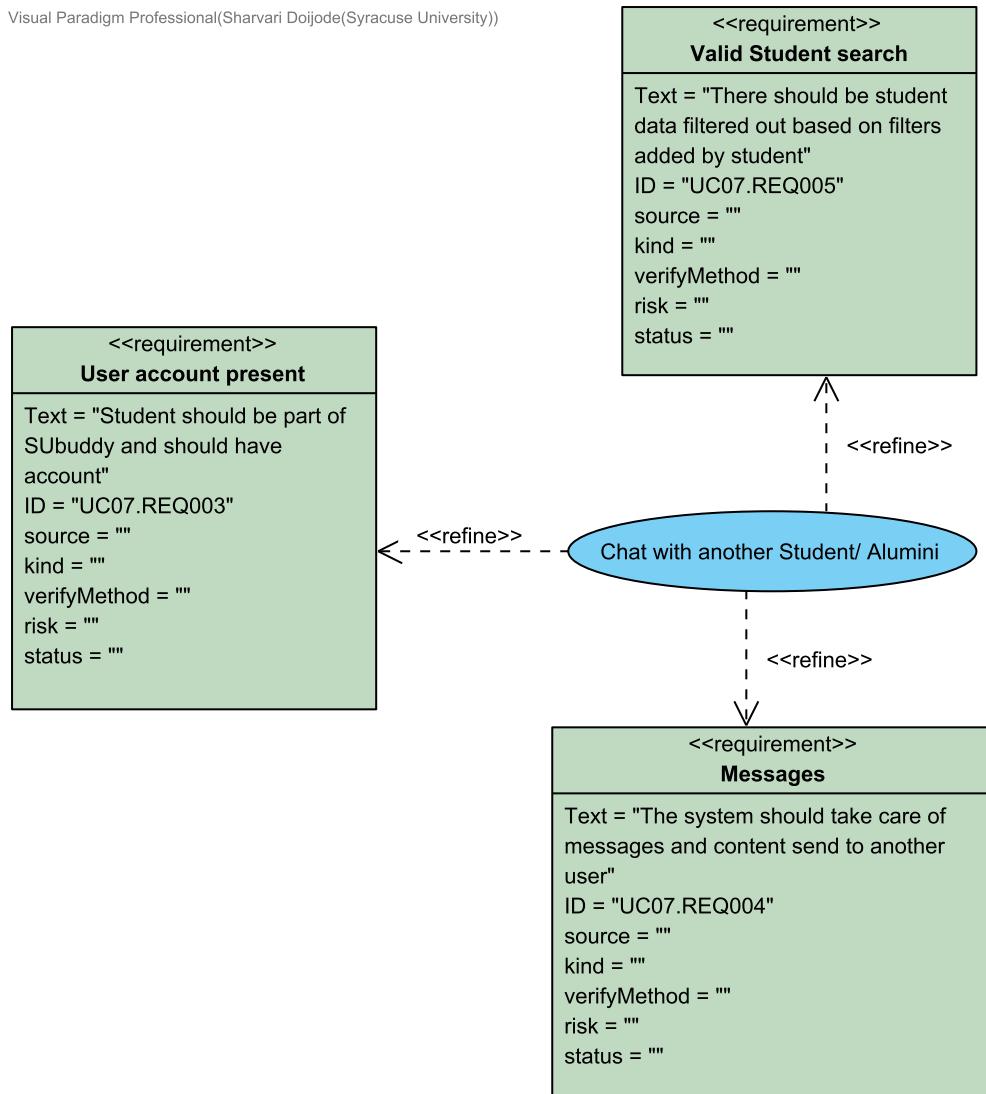
● 2.2.2.17. *Select user to chat with*

2.2.2.17.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.18. Chat with another Student/ Alumni Requirement Diagram

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



2.2.2.19. Chat with another Student/ Alumni

ID: UC07

- 1.Student and Student can communicate with each other.
- 2.Student and Alumni can communicate with each other.

2.2.2.20. User account present

ID: UC07.REQ003

Student should be part of SUbuddy and should have account

2.2.2.21. Messages

ID: UC07.REQ004

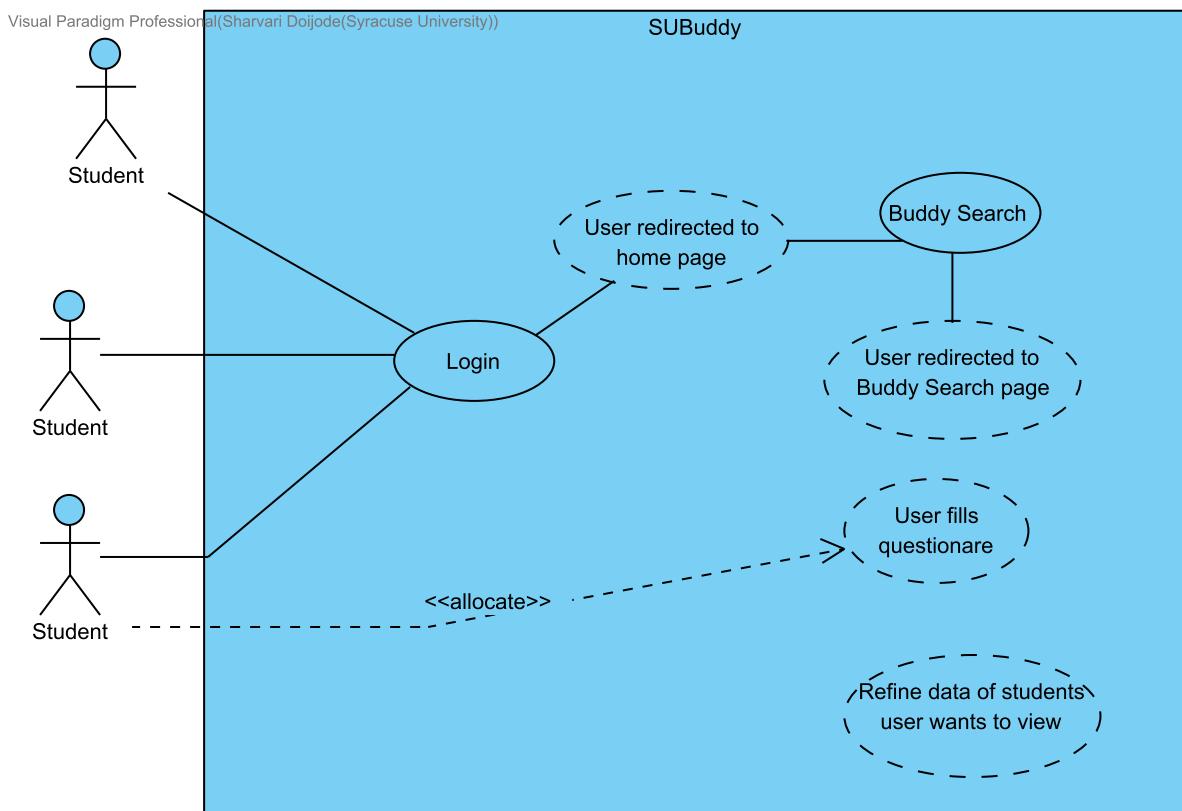
The system should take care of messages and content send to another user

2.2.2.22. Valid Student search

ID: UC07.REQ005

There should be student data filtered out based on filters added by student

2.2.2.23. Buddy /Alumni search



This use case details the functionality that allows users to search for and connect with buddies or alumni. Users can specify search criteria such as location, interests, or study fields, and the system will provide a list of matching profiles. This enables users to build networks and engage with the community based on commonalities and shared experiences.

2.2.2.24. Refine data of students user wants to view

2.2.2.24.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.25. Buddy Search

ID: UC03.UC17

Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A

Assumptions	N/A
--------------------	-----

2.2.2.26. Login

ID: UC30

Primary Actors	Actor2, Actor3, Student
Level	N/A
Complexity	N/A
Use Case Status	N/A
Implementation Status	N/A
Preconditions	N/A
Post-conditions	N/A
Author	N/A
Assumptions	N/A

2.2.2.27. Student

ID: UC03.AC14

2.2.2.27.1. Properties

Abstract	false
Leaf	false
Root	false

2.2.2.27.2. Relationships

Relationship	From	To
---unnamed	Student	User fills questionare
—unnamed	Login	Student
—unnamed	Login	Student
—unnamed	Login	Student

2.2.2.28. Student

ID: UC03.AC14

2.2.2.28.1. Properties

Abstract	false
Leaf	false
Root	false

2.2.2.28.2. Relationships

Relationship	From	To
---unnamed	Student	User fills questionare

Relationship	From	To
— unnamed	 Login	 Student
— unnamed	 Login	 Student
— unnamed	 Login	 Student

2.2.2.29. Student

ID: UC03.AC14

2.2.2.29.1. Properties

Abstract	false
Leaf	false
Root	false

2.2.2.29.2. Relationships

Relationship	From	To
--- unnamed	 Student	 User fills questionare
— unnamed	 Login	 Student
— unnamed	 Login	 Student
— unnamed	 Login	 Student

2.2.2.30. User fills questionare

2.2.2.30.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.30.2. Relationships

Relationship	From	To
--- unnamed	 Student	 User fills questionare

2.2.2.31. User redirected to Buddy Search page

2.2.2.31.1. Properties

Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.31.2. Relationships

Relationship	From	To
—unnamed	 Buddy Search	 User redirected to Buddy Search page

2.2.2.32. User redirected to home page

2.2.2.32.1. Properties

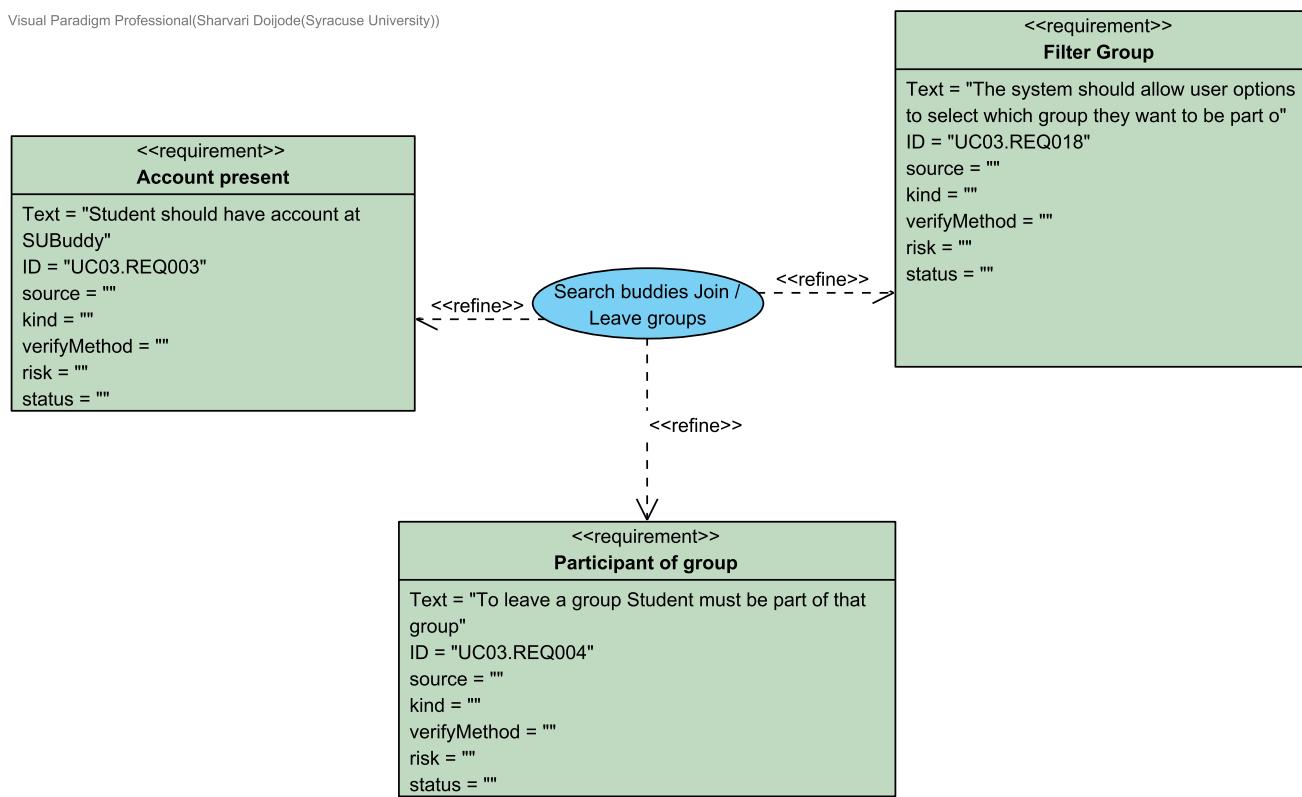
Visibility	Unspecified
Abstract	false
Leaf	false
Root	false

2.2.2.32.2. Relationships

Relationship	From	To
—unnamed	 User redirected to home page	 Buddy Search
—unnamed	 Login	 User redirected to home page

2.2.2.33. Join / Leave groups Requirement Diagram

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



2.2.2.34. Search buddies Join / Leave groups

ID: UC03

Students join the group

2.2.2.35. Account present

ID: UC03.REQ003

Student should have account at SUBuddy

2.2.2.36. Filter Group

ID: UC03.REQ018

The system should allow user options to select which group they want to be part o

2.2.2.37. Participant of group

ID: UC03.REQ004

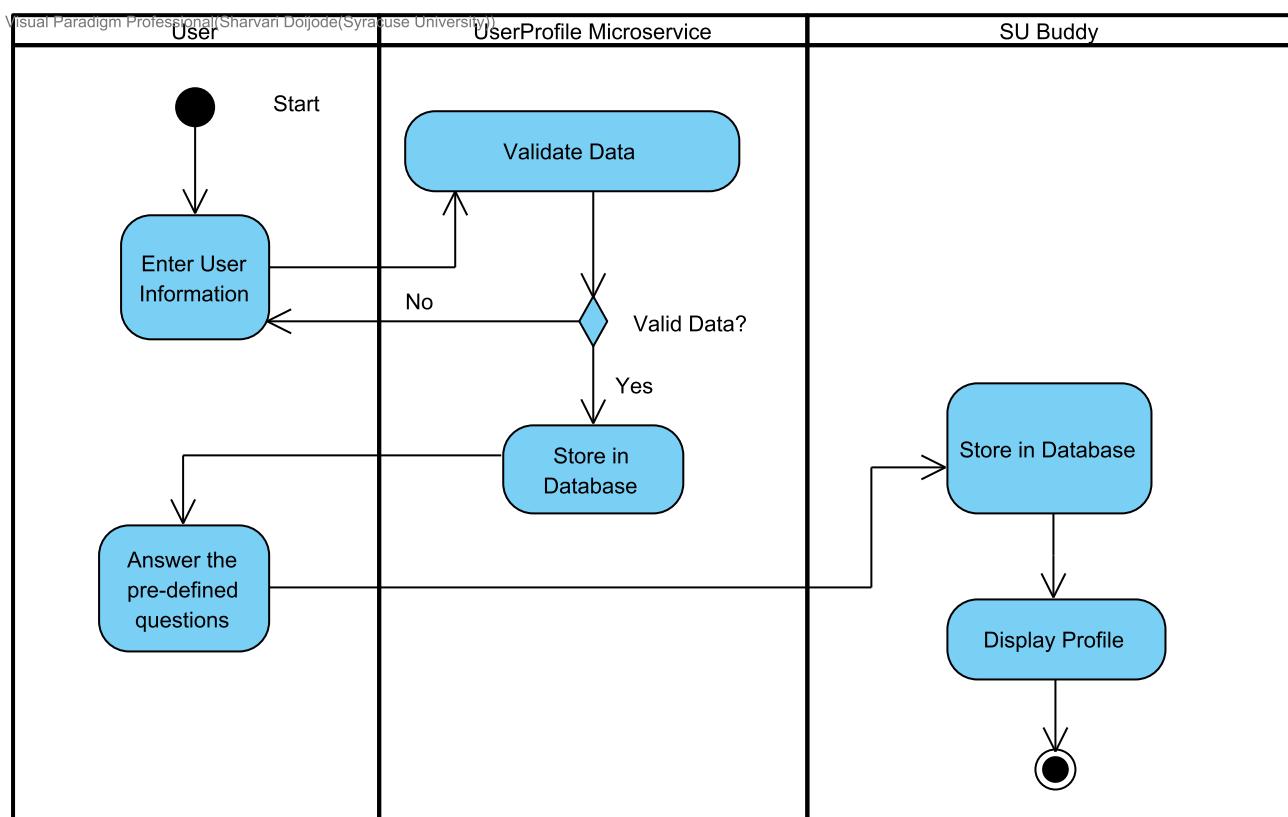
To leave a group Student must be part of that group

3. Specific Requirements

3.1. Overview

This section provides the information regarding the various activities performed by user while user interacts with the use cases.

3.1.1. Signup Activity Diagram



The activity begins with the user entering their personal information, such as name, email, and password, and answering predefined questions which may pertain to their interests or preferences. This data is then sent to the UserProfile Microservice for validation. If the data is not valid, the user is prompted to re-enter the information. Upon successful validation, the

information is stored in the database. The SU Buddy system then takes this information, stores it again for its context (which might involve additional processing or data enrichment), and finally displays the newly created user profile. This end-to-end process is critical for onboarding new users to the SU Buddy platform, ensuring that user data is accurate and profiles are set up correctly for subsequent use.

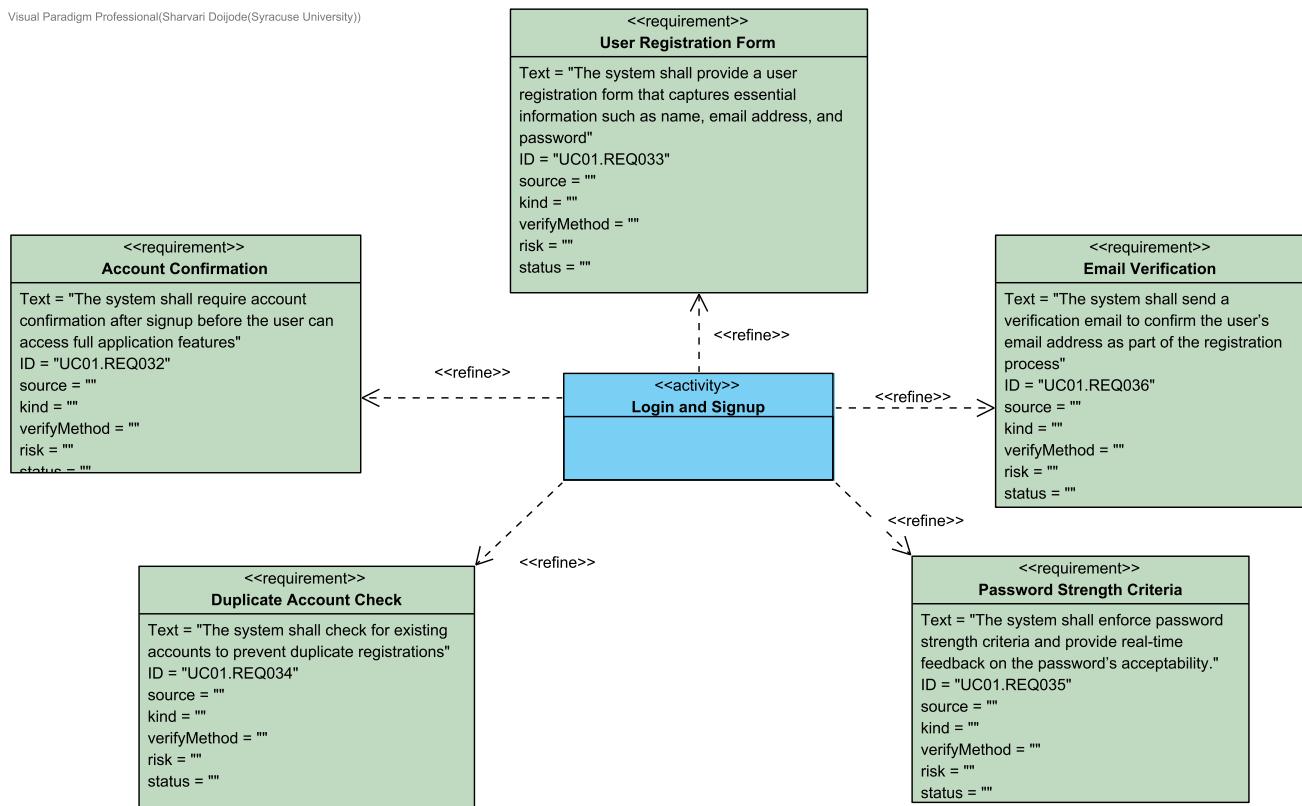
Remember to adjust the specifics of the specifications and descriptions according to the precise details and requirements of your system.

3.1.2. Signup Activity

3.1.2.1. Sub Diagrams

3.1.2.1.1. Signup System Requirement Diagram

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



3.1.3. unnamed

3.1.3.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.3.2. Relationships

Relationship	From	To
unnamed		unnamed

3.1.4. Enter User Information

User enters the necessary information required to create the account

3.1.4.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.4.2. Relationships

Relationship	From	To
CF No	⊕ Valid Data?	⊖ Enter User Information
CF unnamed	⊖ Enter User Information	⊖ Validate Data
CF unnamed	Start	⊖ Enter User Information

3.1.5. Answer the pre-defined questions

User fills the provided questionnaire which will be further used to segregate the students.

3.1.5.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.5.2. Relationships

Relationship	From	To
CF unnamed	⊖ Answer the pre-defined questions	⊖ Store in Database
CF unnamed	⊖ Store in Database	⊖ Answer the pre-defined questions

3.1.6. Display Profile

Once the account is created successfully, the created user profile is displayed to user

3.1.6.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.6.2. Relationships

Relationship	From	To
CF unnamed	⊖ Display Profile	⊕ unnamed

Relationship	From	To
unnamed	Store in Database	Display Profile

3.1.7. Start

Start of the activity

3.1.7.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.7.2. Relationships

Relationship	From	To
unnamed	Start	Enter User Information

3.1.8. Store in Database

After filling the basic questions given to users, the data is stored in database

3.1.8.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.8.2. Relationships

Relationship	From	To
unnamed	Store in Database	Display Profile
unnamed	Answer the pre-defined questions	Store in Database

3.1.9. Store in Database

After data validation data is stored in the database.

3.1.9.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.9.2. Relationships

Relationship	From	To
CF>unnamed	[Store in Database]	[Answer the pre-defined questions]
CF>Yes	[Valid Data?]	[Store in Database]

3.1.10. Valid Data?

Decision node to define further actions based on the result.

3.1.10.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.10.2. Relationships

Relationship	From	To
CF>No	[Valid Data?]	[Enter User Information]
CF>unnamed	[Validate Data]	[Valid Data?]
CF>Yes	[Valid Data?]	[Store in Database]

3.1.11. Validate Data

The data validation takes place here. The email ID would be verified whether its a valid Syracuse University mail.

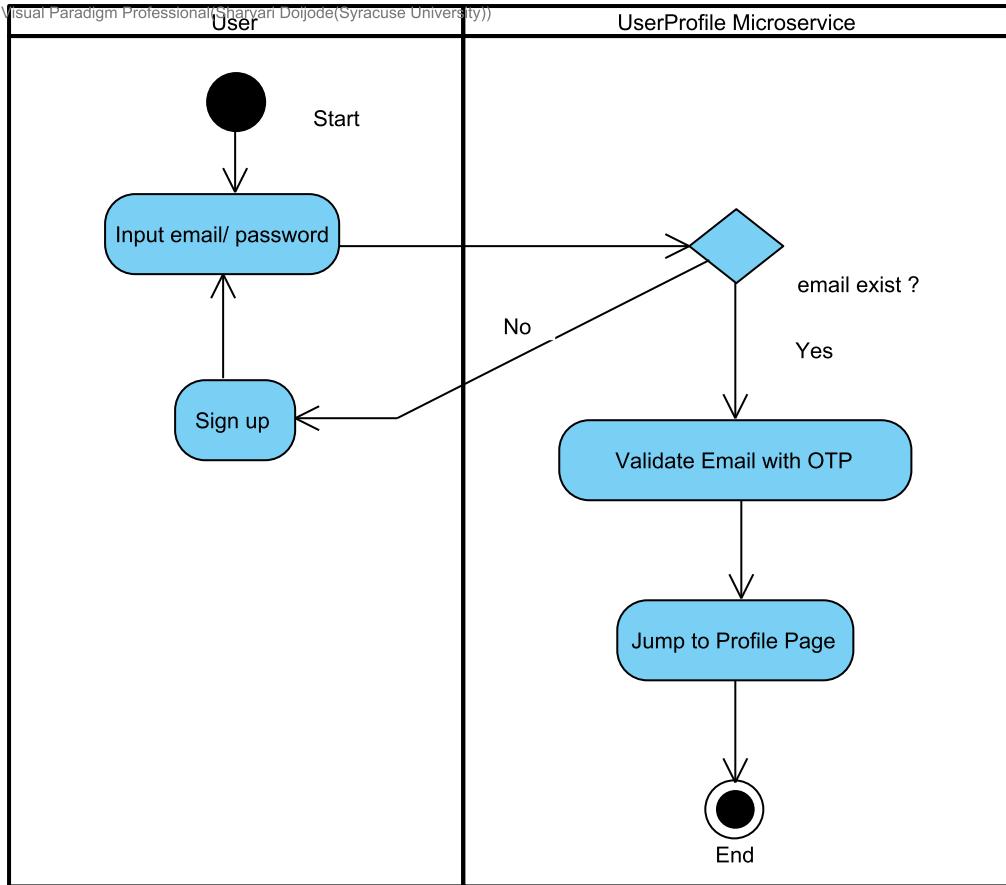
3.1.11.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.11.2. Relationships

Relationship	From	To
CF>unnamed	[Validate Data]	[Valid Data?]
CF>unnamed	[Enter User Information]	[Validate Data]

3.1.12. login Activity Diagram



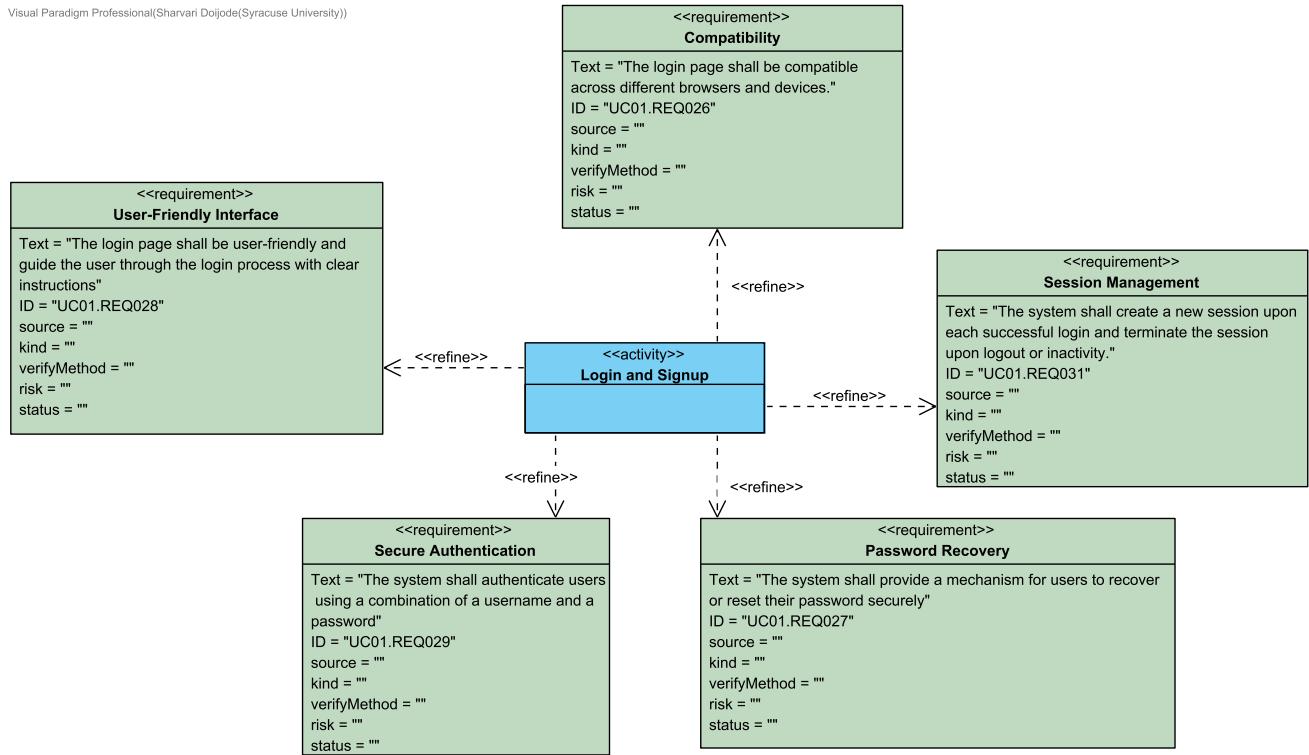
The login workflow begins when the user enters their chosen email and password. After selecting the login option, the UserProfile Microservice takes over to validate the email by sending an OTP to the user's provided email address. The user must enter this OTP to proceed. If the email is verified and does not exist in the system's records, the user can proceed to set up their profile. If the email already exists, implying that the user has an account, the system redirects them to the profile page, perhaps to initiate a password recovery or to log in directly. This workflow is crucial for ensuring that user accounts are secured and tied to a valid email address.

3.1.13. Login

3.1.13.1. Sub Diagrams

3.1.13.1.1. Login System Requirement Diagram

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



3.1.14. End

3.1.14.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.14.2. Relationships

Relationship	From	To
CF unnamed	[] Jump to Profile Page	(End)

3.1.15. Input email/ password

User is prompted to the Syracuse Email ID and the password to access their account.

3.1.15.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.15.2. Relationships

Relationship	From	To
CF unnamed	[] Input email/ password	[] email exist ?

Relationship	From	To
CF unnamed	Sign up	Input email/ password
CF unnamed	Start	Input email/ password

3.1.16. Jump to Profile Page

Once the user is validated, the profile is visible to user

3.1.16.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.16.2. Relationships

Relationship	From	To
CF unnamed	Jump to Profile Page	End
CF unnamed	Validate Email with OTP	Jump to Profile Page

3.1.17. Sign up

Signup page for user if a new user tries to login before even creating the account in SUBuddy

3.1.17.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.17.2. Relationships

Relationship	From	To
CF No	email exist ?	Sign up
CF unnamed	Sign up	Input email/ password

3.1.18. Start

Start of the activity

3.1.18.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.18.2. Relationships

Relationship	From	To
unnamed	Start	Input email/ password

3.1.19. Validate Email with OTP

OTP is sent to the registered mailID and then login attempt made by user will be authenticated.

3.1.19.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.19.2. Relationships

Relationship	From	To
unnamed	Validate Email with OTP	Jump to Profile Page
Yes	email exist ?	Validate Email with OTP

3.1.20. email exist ?

This is the decision node to check if the user with this particular email ID exists in the system. If yes then OTP is sent to that user. If email ID does not exist in system that means user has no account, so he is directed towards signup page again.

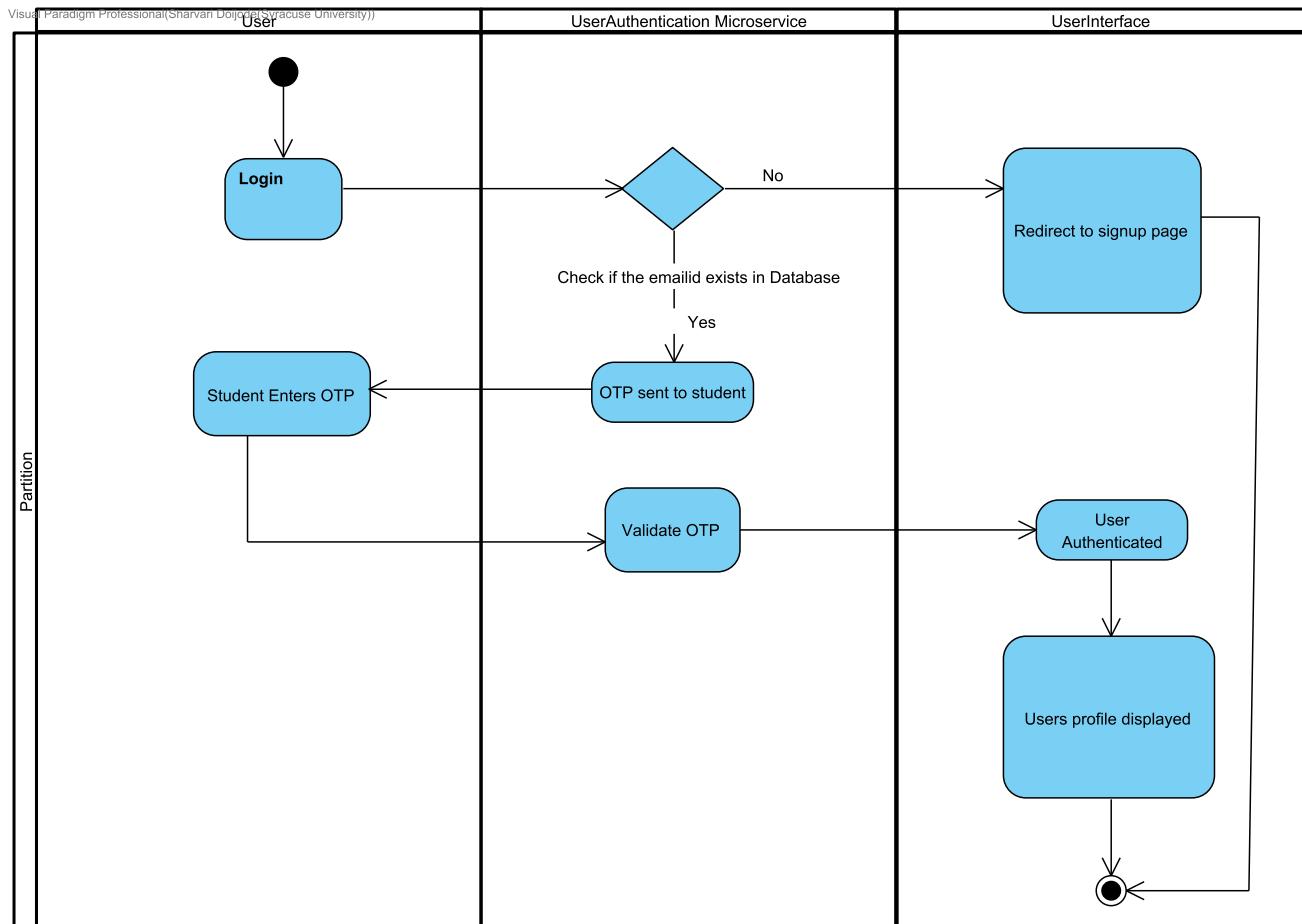
3.1.20.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.20.2. Relationships

Relationship	From	To
No	email exist ?	Sign up
unnamed	Input email/ password	email exist ?
Yes	email exist ?	Validate Email with OTP

3.1.21. User Authentication



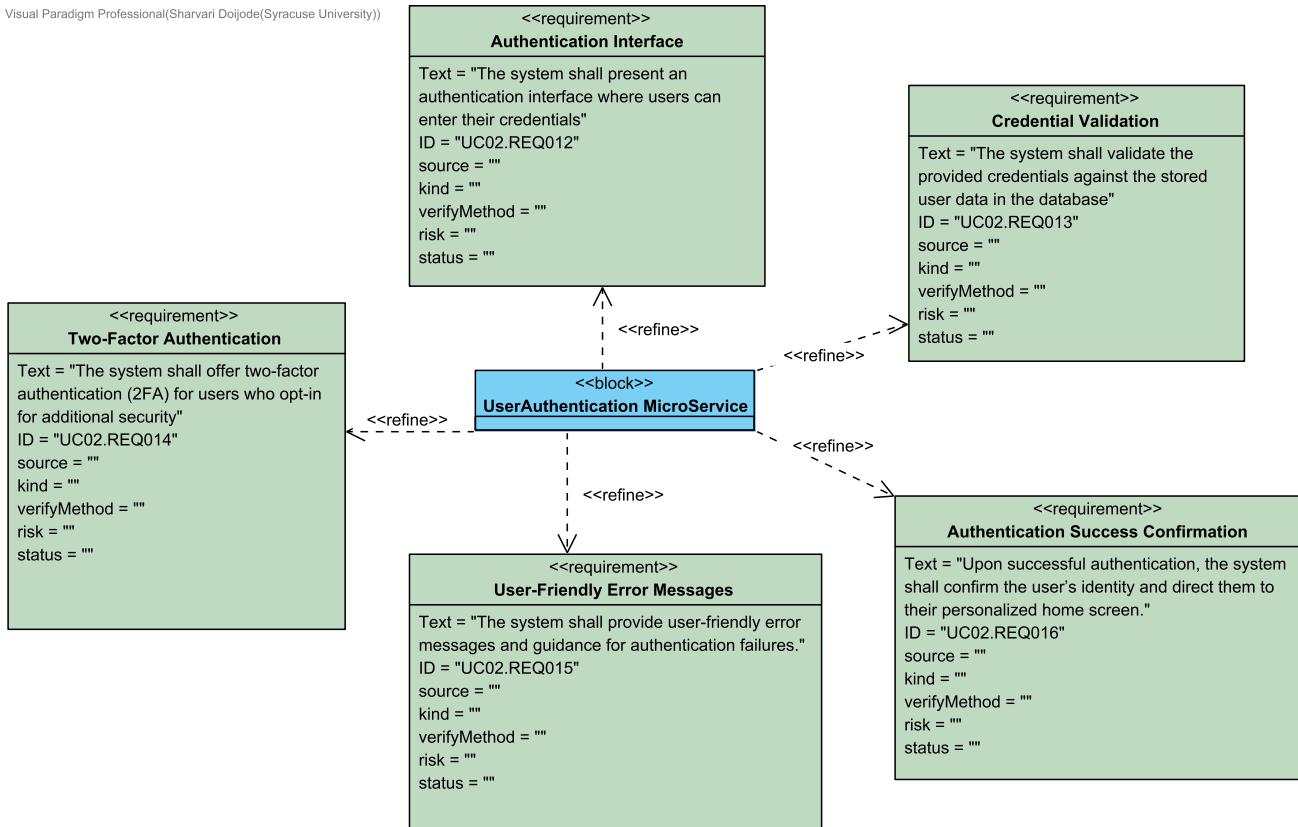
This activity diagram captures the sequence of steps involved in an OTP-based authentication process for a user attempting to access their profile. The process begins with the user logging in, followed by a verification step to determine if the email exists in the database. If the email is not found, the user is redirected to the signup page. If the email exists, an OTP is sent to the user. The user then enters the OTP, which the system validates. Upon successful validation, the user is authenticated, and their profile is displayed.

3.1.22. Authentication

3.1.22.1. Sub Diagrams

3.1.22.1.1. Authentication Requirement Diagram

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



3.1.23. unnamed

3.1.23.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.23.2. Relationships

Relationship	From	To
CF unnamed	[-] Redirect to signup page	○ unnamed
CF unnamed	[-] Users profile displayed	○ unnamed

3.1.24. unnamed

3.1.24.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.24.2. Relationships

Relationship	From	To
CF→unnamed	● unnamed	■ Login

3.1.25. Check if the emailid exists in Database

Decision node to check if the user, already has an account in SUBuddy.

3.1.25.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.25.2. Relationships

Relationship	From	To
CF→No	◆ Check if the emailid exists in Database	■ Redirect to signup page
CF→unnamed	● Login	◆ Check if the emailid exists in Database
CF→Yes	◆ Check if the emailid exists in Database	■ OTP sent to student

■ 3.1.26. Login

Student attempts to login

3.1.26.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.26.2. Relationships

Relationship	From	To
CF→unnamed	■ Login	◆ Check if the emailid exists in Database
CF→unnamed	● unnamed	■ Login

■ 3.1.27. OTP sent to student

User authentication part plays important role here

3.1.27.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.27.2. Relationships

Relationship	From	To
CF→unnamed	→ OTP sent to student	→ Student Enters OTP
CF→Yes	→ Check if the emailid exists in Database	→ OTP sent to student

3.1.28. Redirect to signup page

Since user does not have a profile , user is redirected to signup page to register first

3.1.28.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.28.2. Relationships

Relationship	From	To
CF→No	→ Check if the emailid exists in Database	→ Redirect to signup page
CF→unnamed	→ Redirect to signup page	○ unnamed

3.1.29. Student Enters OTP

As a part of Authentication, student receives a OTP

3.1.29.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.29.2. Relationships

Relationship	From	To
CF→unnamed	→ Student Enters OTP	→ Validate OTP
CF→unnamed	→ OTP sent to student	→ Student Enters OTP

3.1.30. User Authenticated

The details and

3.1.30.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.30.2. Relationships

Relationship	From	To
 unnamed	 User Authenticated	 Users profile displayed
 unnamed	 Validate OTP	 User Authenticated

3.1.31. Users profile displayed

After successful authentication, user profile is displayed and student is logged in to his account in SU buddy.

3.1.31.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.31.2. Relationships

Relationship	From	To
 unnamed	 Users profile displayed	 unnamed
 unnamed	 User Authenticated	 Users profile displayed

3.1.32. Validate OTP

The OTP entered by user and the OTP sent is matched and authentication process takes place

3.1.32.1. Properties

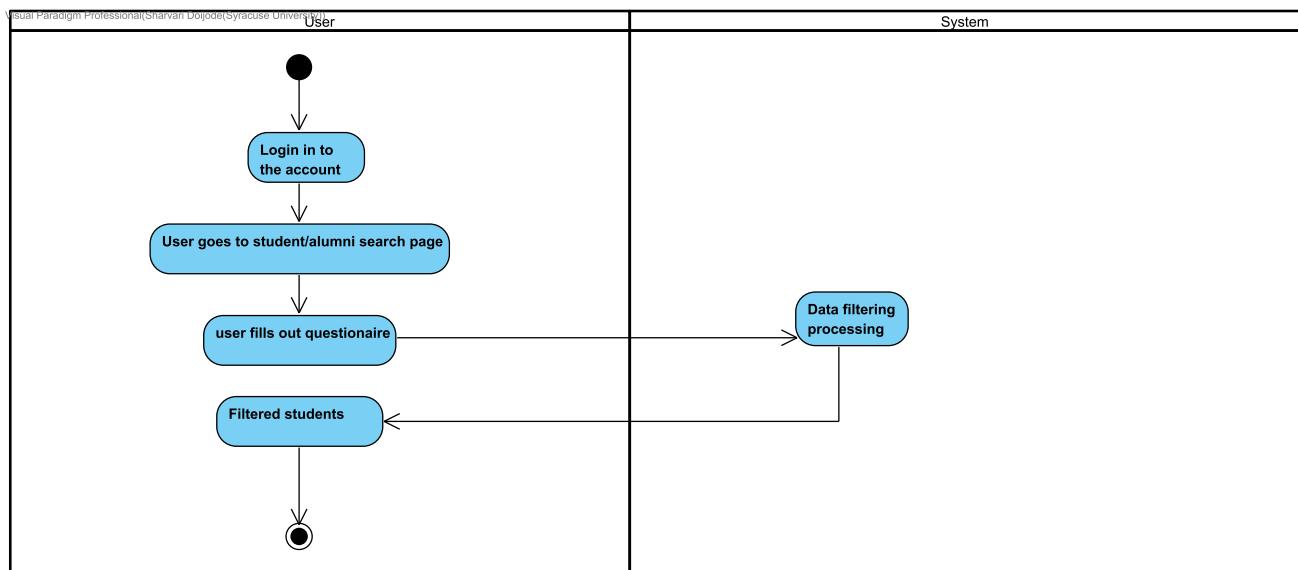
Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.32.2. Relationships

Relationship	From	To
 unnamed	 Validate OTP	 User Authenticated

Relationship	From	To
CF unnamed	Student Enters OTP	Validate OTP

3.1.33. Buddy/Alumni search



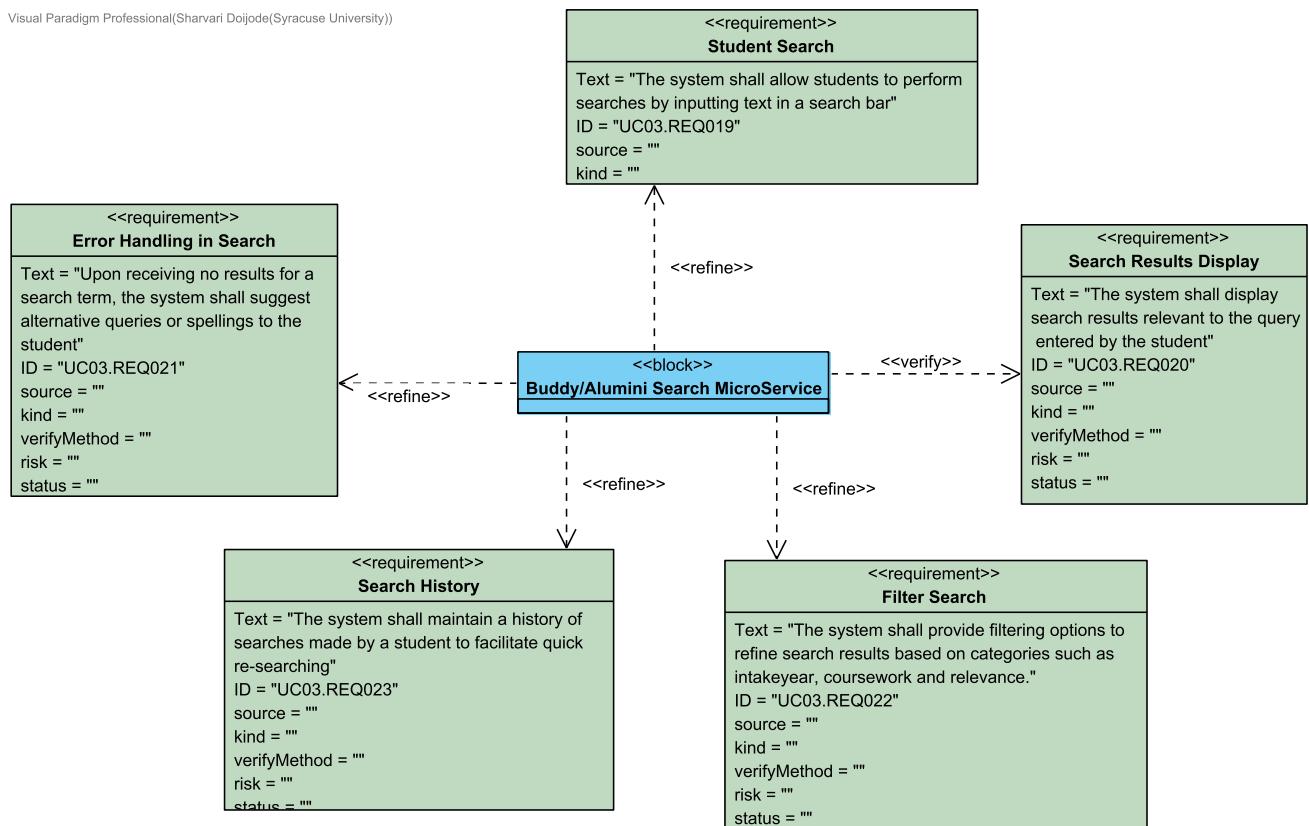
Upon entering the system, the user is presented with the homepage after a successful login. The user's journey to finding a student or alumni begins when they navigate to the dedicated search page, where they are prompted to fill out a detailed questionnaire to tailor the search criteria. The system takes these inputs and processes them through its data filtering mechanism. It intelligently curates a list of profiles that match the user's interests or search parameters. These profiles are then displayed in a card format for the user to view, allowing them to explore connections and engage with other individuals on the platform. This process is designed to be intuitive and user-friendly, ensuring users can effectively search for and connect with the student or alumni network.

3.1.34. Buddy Alumni Search

3.1.34.1. Sub Diagrams

3.1.34.1.1. Search System Required Diagram

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



3.1.35. unnamed

3.1.35.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.35.2. Relationships

Relationship	From	To
CF unnamed	Filtered students	unnamed

3.1.36. unnamed

3.1.36.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.36.2. Relationships

Relationship	From	To
CF unnamed	unnamed	Login in to the account

3.1.37. Data filtering processing

Based on the responses provided by user, the data is filtered and fetched from the Database and only those students profile are sent back to user

3.1.37.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.37.2. Relationships

Relationship	From	To
 unnamed	 Data filtering processing	 Filtered students
 unnamed	 user fills out questionaire	 Data filtering processing

3.1.38. Filtered students

This page displays the refined search based on users criteria and responses to the questions asked before.

3.1.38.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.38.2. Relationships

Relationship	From	To
 unnamed	 Filtered students	 unnamed
 unnamed	 Data filtering processing	 Filtered students

3.1.39. Login in to the account

User logs in and is redirected to homepage

3.1.39.1. Properties

Language	N/A
Precondition	N/A

Postcondition	N/A
Body	User logs in and is redirected to homepage
Single execution	false
Read only	false
Re-entrant	true

3.1.39.2. Relationships

Relationship	From	To
unnamed	Login in to the account	User goes to student/alumni search page
unnamed	unnamed	Login in to the account

3.1.40. User goes to student/alumni search page

The user is provided with the home page which has all the other student profiles available in the system.

3.1.40.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.40.2. Relationships

Relationship	From	To
unnamed	User goes to student/alumni search page	user fills out questionnaire
unnamed	Login in to the account	User goes to student/alumni search page

3.1.41. user fills out questionnaire

Option for the user to filter out the search. Questions provided for segregating the data

3.1.41.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false

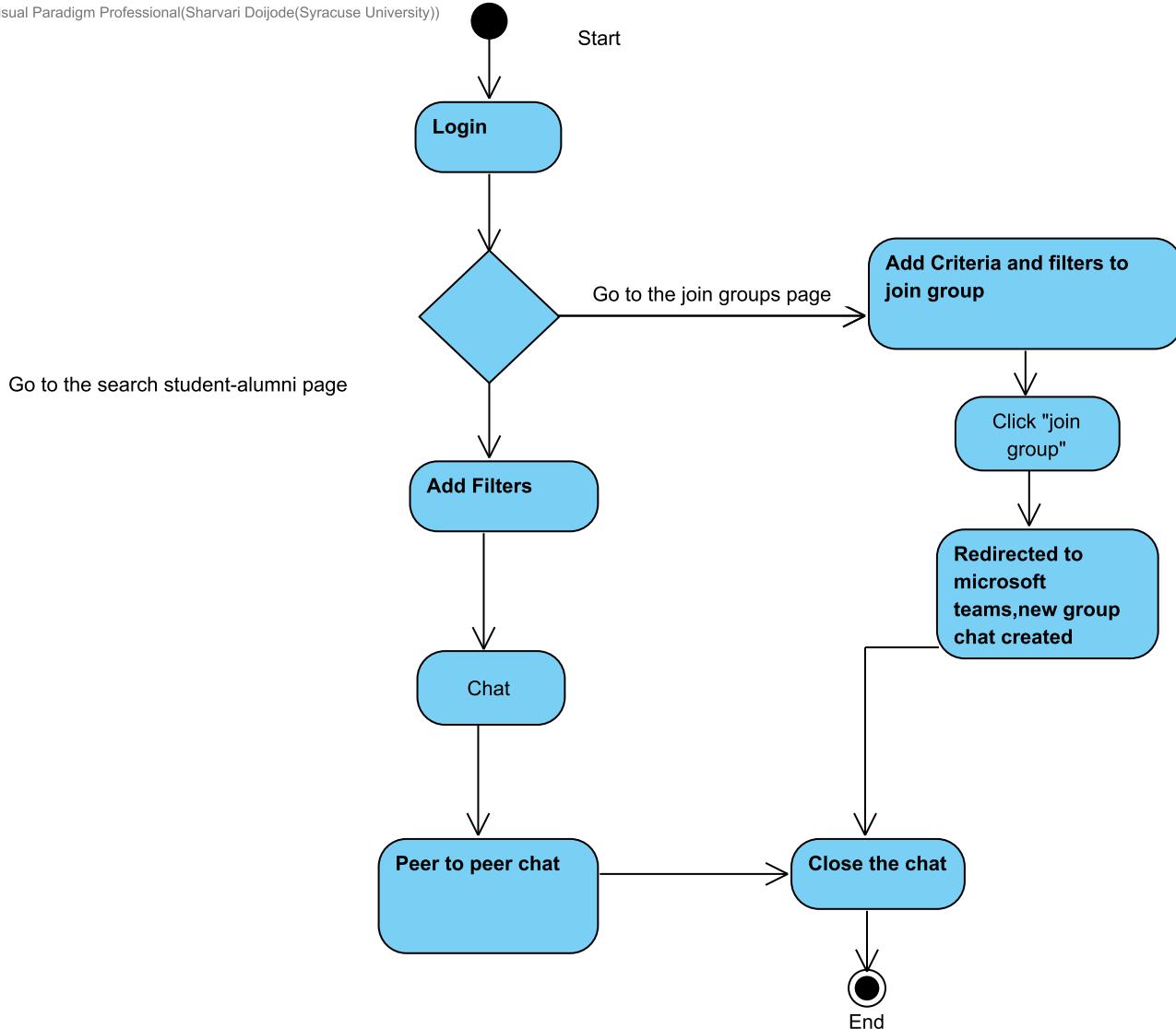
Re-entrant	true
------------	------

3.1.41.2. Relationships

Relationship	From	To
unnamed	user fills out questionnaire	Data filtering processing
unnamed	User goes to student/alumni search page	user fills out questionnaire

3.1.42. Chat services

Visual Paradigm Professional(Sharvari Doijode(Syracuse University))



This workflow illustrates the process for a user engaging in chat activities within a platform that integrates with Microsoft Teams. The user begins by logging into the system, after which they are directed to the student-alumni search page. Here, the user can add filters to refine their search for other users to chat with. Once the filters are applied, the user can initiate a chat. Alternatively, the user can navigate to the join groups page, where they can add criteria and filters to select a group. Upon clicking "join group," the user is redirected to Microsoft Teams, and a new group chat is created. After engaging in peer-to-peer chat, the user can choose to close the chat, which concludes the session. This integration provides a seamless transition from the platform's internal chat feature to a group chat environment facilitated by Microsoft Teams, fostering collaboration and connectivity among users.

3.1.43. unnamed

Decision node to check if user wants to chat Individualy or join a group

3.1.43.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.43.2. Relationships

Relationship	From	To
CF Go to the join groups page	↳ unnamed	[-] Add Criteria and filters to join group
CF Go to the search student-alumni page	↳ unnamed	[-] Add Filters
CF unnamed	↳ unnamed	[-] Add Criteria and filters to join group
CF unnamed	[-] Login	↳ unnamed

3.1.44. Add Criteria and filters to join group

Select a bunch of parameters concerning the group you want to join

3.1.44.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.44.2. Relationships

Relationship	From	To
CF Go to the join groups page	↳ unnamed	[-] Add Criteria and filters to join group
CF unnamed	[-] Add Criteria and filters to join group	[-] Click "join group"
CF unnamed	↳ unnamed	[-] Add Criteria and filters to join group

3.1.45. Add Filters

Select a bunch of parameters to find students/alumnis based on which the student profiles would be displayed to user

3.1.45.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.45.2. Relationships

Relationship	From	To
Go to the search student-alumni page	unnamed	Add Filters
unnamed	Add Filters	Chat

3.1.46. Chat

Chat with the desired student.

3.1.46.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.46.2. Relationships

Relationship	From	To
unnamed	Chat	Peer to peer chat
unnamed	Add Filters	Chat

3.1.47. Click "join group"

Proceed to Join group

3.1.47.1. Properties

Effect	N/A
Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.47.2. Relationships

Relationship	From	To
unnamed	Click "join group"	Redirected to microsoft teams, new group chat created

Relationship	From	To
unnamed	Add Criteria and filters to join group up	Click "join group"

3.1.48. Close the chat

3.1.48.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.48.2. Relationships

Relationship	From	To
unnamed	Close the chat	End
unnamed	Redirected to microsoft teams, new group chat created	Close the chat
unnamed	Peer to peer chat	Close the chat

3.1.49. End

3.1.49.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.49.2. Relationships

Relationship	From	To
unnamed	Close the chat	End

3.1.50. Login

User logs in and is redirected to the homepage

3.1.50.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A

Single execution	false
Read only	false
Re-entrant	true

3.1.50.2. Relationships

Relationship	From	To
unnamed		unnamed
unnamed		

3.1.51. Peer to peer chat

Individual interaction provided to users who want to just connect with peers.

3.1.51.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

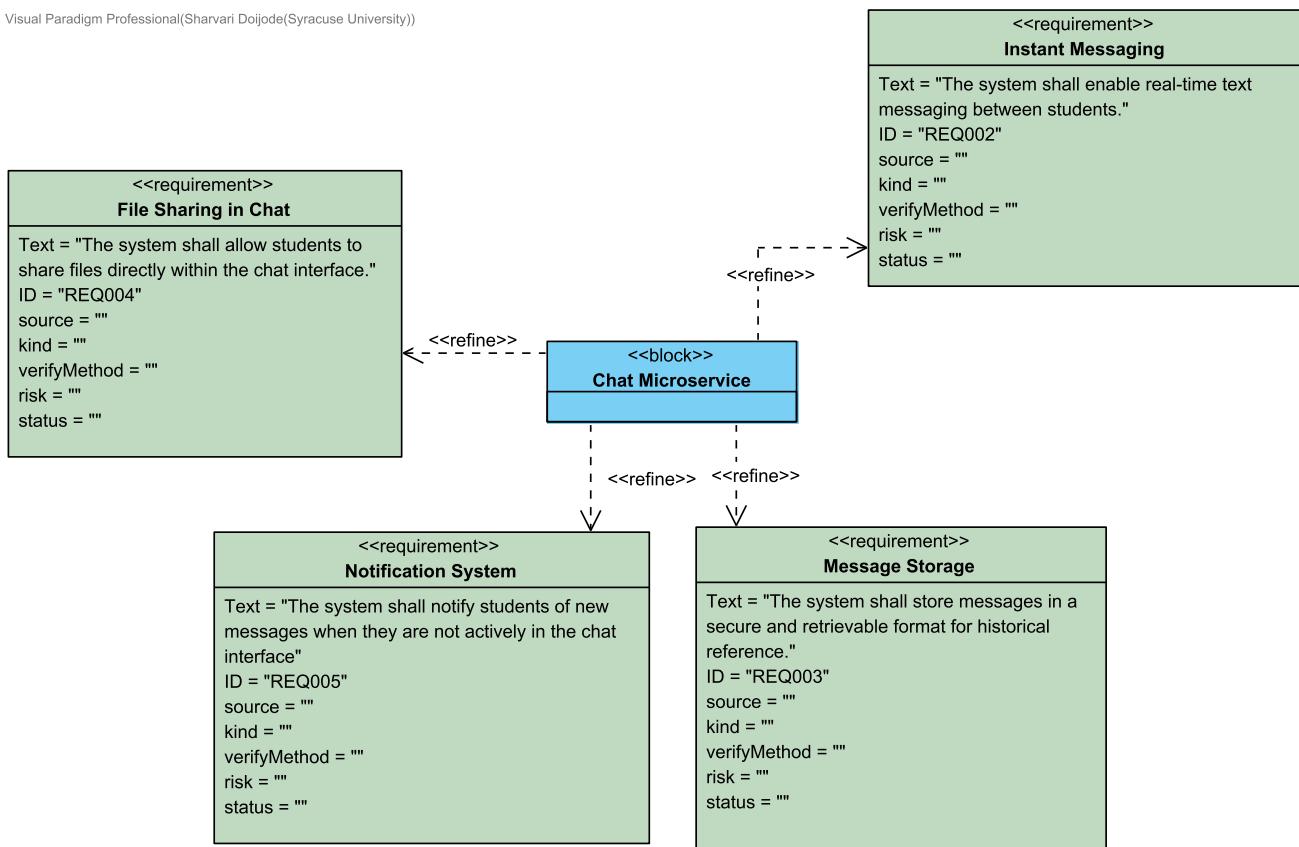
3.1.51.2. Relationships

Relationship	From	To
unnamed		
unnamed		

3.1.51.3. Sub Diagrams

3.1.51.3.1. Peer to peer chat System Requirement Diagram

Visual Paradigm Professional(Sharvari Dojode(Syracuse University))



3.1.52. Redirected to microsoft teams,new group chat created

User would be directed to Microsoft teams and added to the required group student want to join

3.1.52.1. Properties

Language	N/A
Precondition	N/A
Postcondition	N/A
Body	N/A
Single execution	false
Read only	false
Re-entrant	true

3.1.52.2. Relationships

Relationship	From	To
CF unnamed	[] Redirected to microsoft teams, new group chat created	[] Close the chat
CF unnamed	[] Click "join group"	[] Redirected to microsoft teams, new group chat created

3.1.53. Start

3.1.53.1. Properties

Visibility	Unspecified
Must isolate	false
Leaf	false

3.1.53.2. Relationships

Relationship	From	To
CF unnamed	 Start	 Login

3.2. Performance Requirement

3.2.1. Login and Signup

3.2.2. Compatibility

ID: UC01.REQ026

The login page shall be compatible across different browsers and devices.

3.2.3. Password Recovery

ID: UC01.REQ027

The system shall provide a mechanism for users to recover or reset their password securely

3.2.4. Secure Authentication

ID: UC01.REQ029

The system shall authenticate users using a combination of a username and a password

3.2.5. Session Management

ID: UC01.REQ031

The system shall create a new session upon each successful login and terminate the session upon logout or inactivity.

3.2.6. User-Friendly Interface

ID: UC01.REQ028

The login page shall be user-friendly and guide the user through the login process with clear instructions

3.2.7. Login and Signup

3.2.8. Account Confirmation

ID: UC01.REQ032

The system shall require account confirmation after signup before the user can access full application features

3.2.9. Duplicate Account Check

ID: UC01.REQ034

The system shall check for existing accounts to prevent duplicate registrations

3.2.10. Email Verification

ID: UC01.REQ036

The system shall send a verification email to confirm the user's email address as part of the registration process

3.2.11. Password Strength Criteria

ID: UC01.REQ035

The system shall enforce password strength criteria and provide real-time feedback on the password's acceptability.

3.2.12. User Registration Form

ID: UC01.REQ033

The system shall provide a user registration form that captures essential information such as name, email address, and password

3.2.13. UserAuthentication MicroService

Authenticates user credentials against the database records, manages active sessions, and issues authentication tokens for session management.

3.2.14. User-Friendly Error Messages

ID: UC02.REQ015

The system shall provide user-friendly error messages and guidance for authentication failures.

3.2.15. Authentication Interface

ID: UC02.REQ012

The system shall present an authentication interface where users can enter their credentials

3.2.16. Authentication Success Confirmation

ID: UC02.REQ016

Upon successful authentication, the system shall confirm the user's identity and direct them to their personalized home screen.

3.2.17. Credential Validation

ID: UC02.REQ013

The system shall validate the provided credentials against the stored user data in the database

3.2.18. Two-Factor Authentication

ID: UC02.REQ014

The system shall offer two-factor authentication (2FA) for users who opt-in for additional security

3.2.19. Buddy/Alumni Search MicroService

Enables users to find and connect with other alumni or buddies using search criteria. Integrates with the user profile database to retrieve and filter search results.

3.2.20. Error Handling in Search

ID: UC03.REQ021

Upon receiving no results for a search term, the system shall suggest alternative queries or spellings to the student

3.2.21. Filter Search

ID: UC03.REQ022

The system shall provide filtering options to refine search results based on categories such as intakeyear, coursework and relevance.

3.2.22. Search History

ID: UC03.REQ023

The system shall maintain a history of searches made by a student to facilitate quick re-searching

3.2.23. Search Results Display

ID: UC03.REQ020

The system shall display search results relevant to the query entered by the student

3.2.24. Student Search

ID: UC03.REQ019

The system shall allow students to perform searches by inputting text in a search bar

3.2.25. Chat Microservice

Manages real-time messaging between users, storing and retrieving message histories, and supports notifications for new messages

3.2.26. File Sharing in Chat

ID: REQ004

The system shall allow students to share files directly within the chat interface.

3.2.27. Instant Messaging

ID: REQ002

The system shall enable real-time text messaging between students.

3.2.28. Message Storage

ID: REQ003

The system shall store messages in a secure and retrievable format for historical reference.

3.2.29. Notification System

ID: REQ005

The system shall notify students of new messages when they are not actively in the chat interface

3.2.30. SUBuddyInteraction

Central system block that orchestrates the interaction between various microservices involved in the SUbuddy platform, including user registration, authentication, search, and chat functionalities.

Responsibilities:

1. Coordinate microservices interaction.
2. Ensure data consistency across services.
3. Provide a unified interface for the front-end to interact with backend services.

3.2.31. Buddy/Alumni Search Microservice:

ID: REQ009

The service shall return search results within 3 seconds for standard queries.

3.2.32. Chat Microservice:

ID: REQ007

The service shall have the capability to store and retrieve at least one year's worth of chat history with a response time of less than 5 seconds

3.2.33. UserAuthentication Microservice

ID: REQ008

The service shall authenticate user credentials within 1 second under standard load conditions

3.2.34. UserRegistration Microservice

ID: REQ006

The service shall process registration requests within 2 seconds under standard load conditions

3.3. Design Constraints

3.3.1. SBUddyInteraction

Central system block that orchestrates the interaction between various microservices involved in the SBUddy platform, including user registration, authentication, search, and chat functionalities.

Responsibilities:

1. Coordinate microservices interaction.
2. Ensure data consistency across services.
3. Provide a unified interface for the front-end to interact with backend services.

3.3.2. Buddy/Alumni Search Microservice:

ID: REQ012

Search results must prioritize user privacy and consent, not revealing personal contact information without user permission

3.3.3. Chat Microservice:

ID: REQ013

Must include the ability to archive chats for a minimum duration as per university policy or legal requirements.

3.3.4. UserAuthentication Microservice:

ID: REQ010

The authentication service shall implement rate-limiting to prevent brute force attacks.

3.3.5. UserRegistration Microservice:

ID: REQ011

The registration process must be compatible with the university's student information system for verification purposes.