

# ICDE-[Skill Stack]

Name: Ajay Kumar Lakshmipura Vijaykumar  
Student ID:40168544  
Program: MEng Electrical and Computer Engineering  
Email: [a\\_laks@encs.concordia.ca](mailto:a_laks@encs.concordia.ca)  
SE Background: Familiar with C, Python and OOPS Concepts

Name: Somesh Vemula  
Student ID:40161259  
Program: MEng Electrical and Computer Engineering  
Email: [s\\_emula@encs.concordia.ca](mailto:s_emula@encs.concordia.ca)  
SE Background: Familiar with C, Python and OOPS Concepts

Name: Goutham Kumar Prabhakaran  
Student ID:40160167  
Program: MEng Electrical and Computer Engineering  
Email: [g\\_prabh@encs.concordia.ca](mailto:g_prabh@encs.concordia.ca)  
SE Background: Familiar with C, Python and OOPS Concepts

## I. TOOLS DECIDED

Description	Tool
Code Repository	GitHub
IDE	PyCharm
Programming Language	Python
Group Communication Software	Microsoft Teams
Progress Tracking Tool	JIRA
Database	MongoDB

Table 1: Tools

## II. PROBLEM STATEMENT

### ICDE - [Skill Stack]

The main aim of the project is to develop a course recommendation system, making it easy for a user to choose course from most accessed courses for a particular discipline.

## III. ARCHITECTURE DIAGRAM

### A. High Level Architecture Diagram:

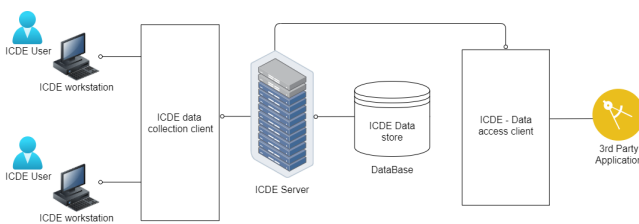


Fig 1.1 ICDE - [Skill Stack] High Level Architecture

Fig.1.1 shows the concrete context of how data flows from multiple users to database and then to third party applications using ICDE - [Skill Stack].

### B. System Level Architecture Diagram:

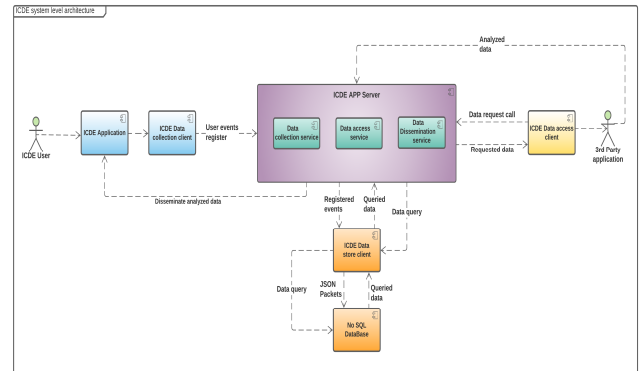


Fig 1.2 ICDE - [Skill Stack] System level Architecture

The overview of the system level architecture is as shown in the above figure (Fig.1.2). This entire application will be hosted in an application server. ICDE data collection client is a part of ICDE application which receives data from ICDE users using the ICDE application. The entire collected data will be sent to third-party application, where the application will further analyze the data for dissemination.

## IV. KEY FUNCTIONALITIES

- Collection of user data**  
Accumulating piles of data provided by the user and storing it.
- Course recommendation based on user's interest**  
From the user data, suggesting the courses based upon the particular interest and help user develop skills in the specific discipline.
- Tracking user actions**  
To make the user engaged, tracking the user actions and presenting it to the user.
- Updating the user profile and storing in Database**  
Storing these user data and preferences in our database handled by NoSQL Database.
- Analyzing the actions**  
Analyzing the various activities done by the user and storing it in the database for analyzing later.

- **Presenting the analyzed data**

Parsing the analyzed data and delivering it statistically with the help of a 3<sup>rd</sup> party application.

## V. USE CASE DIAGRAM

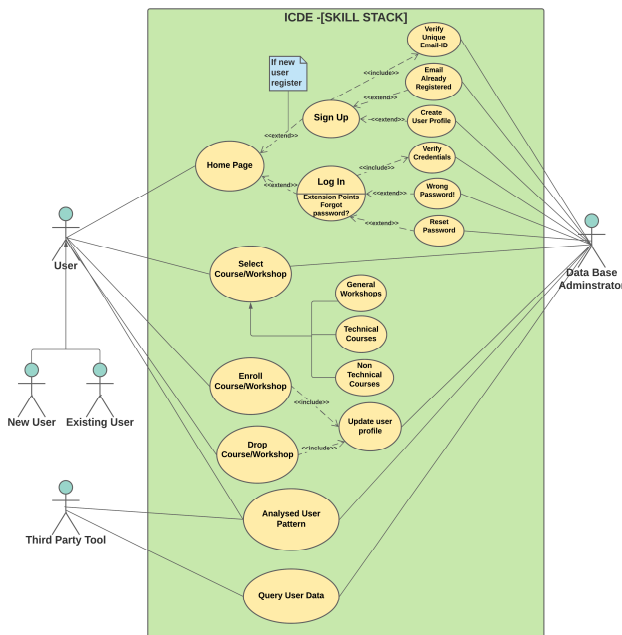


Fig 2. ICDE-Use case Diagram

## VI. USER STORIES

### Home Page

#### Use case - Home Page

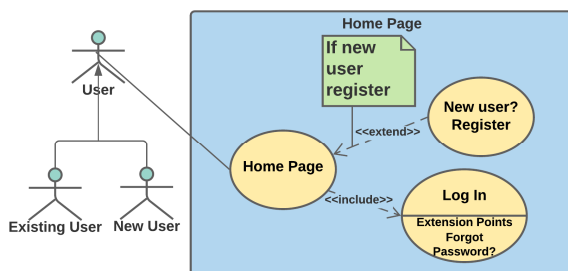


Fig 2.1 Home page use case diagram

**User Story** - He/ She is the new user and needs to create an account, the fields that are displayed "Login", "Sign Up" and "Exit". If the user selects "Login", the system navigates to the login page where it asks for the user credentials. Rather if the user selects "Sign Up", the system asks the user for the details of the user for purpose of registration. If the user selects "Exit", the system exits the application.

**Requirement Specification** - In home page, ICDE-Skill Stack System shall have "Sign Up", "Sign In", "Exit" fields. Upon user selection the ICDE-Skill Stack System shall navigate to Sign-up page and Sign-In respectively.

#### Sub-Requirements –

**SRS\_0101** ICDE-Skill Stack System home page shall have "Sign Up", "Sign In", "Exit" options.

**SRS\_0102** Upon selection, ICDE-Skill Stack System shall navigate to Sign-up page, Sign-In page and Exit the application respectively.

#### Tasks-

**SRT\_0101** **Create home page with "Sign Up" and "Sign-In" options.**

The homepage containing "Sign Up and Sign-In" options is created.

**SRT\_0102** **Navigate to the respective pages.**

On selection in the homepage, the system lands on the respective pages in accordance to the selection.

### Sign Up

#### Use case - Sign Up

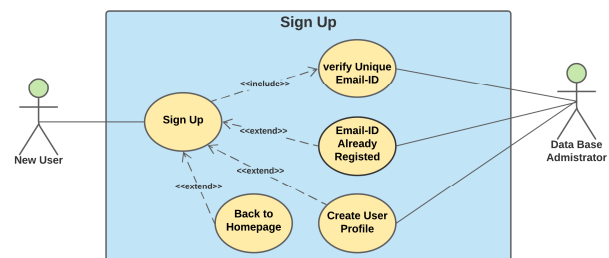


Fig 2.2 Sign Up page use case diagram

**User Story** - He/ She is the new user and needs to create an account, the "Sign Up" page displays the fields to collect the details of the user. The user provides all the details such as, "First Name, Last Name, E-Mail, Password, Confirm Password, Security Questions, Mobile Number" and once he clicks "Submit", the account is created. If the user selects "Back to Homepage", he is navigated to the homepage.

**Requirement Specification** - In registration page, ICDE-Skill Stack System shall collect the user info which are listed below: First Name, Last Name, User Mail ID, Password, Confirm Password, Mobile Number, Discipline. Also, ICDE-Skill Stack System should verify provided User Mail ID is a unique mail id and is not registered before.

#### Sub-Requirements –

**SRS\_0201** Registration page shall have fields listed below that are to be collect from a new user: First Name, Last Name, User Mail ID, Password, Confirm Password, Mobile Number, Discipline.

Also, registration page should have "Back to Homepage" option which enables user to return to home page.

**SRS\_0202** ICDE-Skill Stack System shall verify for unique mail id upon user input.

**SRS\_0203** ICDE-Skill Stack System shall create user profile and store user data in it.

**SRS\_0204** ICDE-Skill Stack System shall upload user profile to data base.

## Tasks-

- SRT\_0201\_01 Create registration page.**  
The registration page is created to collect the information from the user.
- SRT\_0201\_02 Collect user info.**  
User information for the fields displayed in the registration page will be collected.
- SRT\_0201\_03 Add “Back to Homepage” option.**  
In the registration page, “Back to Homepage” option is created for navigating the user to the application homepage.
- SRT\_0202\_01 Validate user Email.**  
The Email entered is validated for its uniqueness.
- SRT\_0202\_02 Password hashing.**  
The password entered is hashed and stored for the purpose of security
- SRT\_0203\_01 Create user profile.**  
The profile is created upon collecting all the information from the user.
- SRT\_0204\_01 Store in database**  
The user information collected is stored in the database.

Also, Login page should have “Back to Homepage” option which enables user to return to home page.

- SRS\_0302** ICDE-Skill Stack System, shall verify the entered credentials and allow user to login or display login.

## Tasks:

- SRT\_0301\_01 Create a login page with login button.**  
The login page which contains the login button is created.
- SRT\_0301\_02 Collect user credentials.**  
The user credentials are collected from the user for logging in.
- SRT\_0301\_03 Add “Back to Homepage” option.**  
“Back to Homepage” is added in order to navigate the user to the homepage from the login page.
- SRT\_0302\_01 Validate user credentials.**  
The user credentials given by the user is validated with the data from database.
- SRT\_0302\_02 Navigate to application main page.**  
The user is navigated to the application main page upon validation of the credentials.

## Log In

### Use case - Log in

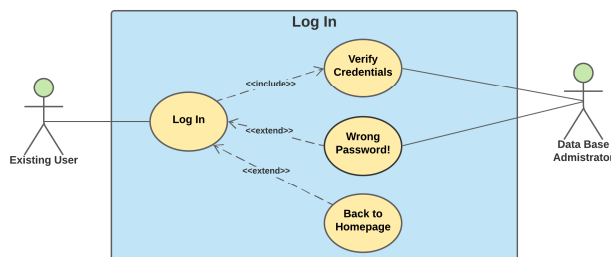


Fig 2.3 Log In page use case diagram

**User Story-** He/ She is a returning user who wish to access the courses, the Login page displays the fields to collect the user credentials. The user provides the user credentials such as, "E-Mail, Password" and once he clicks "Login", the user is logged in. If the user selects "Back to Homepage", he is navigated to the homepage.

**Requirement Specification** - Login page shall have Email and Password fields to fill in by the user. Upon user input ICDE-Skill Stack System verifies the entered credentials and allow user to login or display login error due to wrong password.

Also, Login page should have “Back to Homepage” option which enables user to return to home page.

### Sub-Requirements –

- SRS\_0301** Login page of ICDE-Skill Stack System, shall have “E-Mail” and “Password” fields to enter by existing user. Here Username should be user mail id.

## Reset the password

### Use case - Reset the password

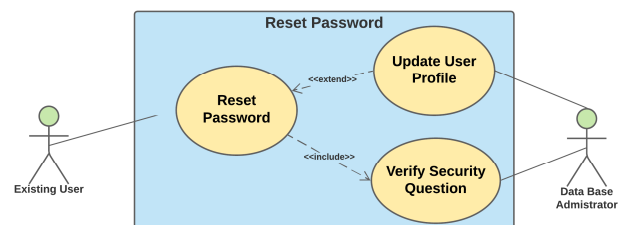


Fig 2.4 Reset Password page use case diagram

**User Story** – On account of forgetting the password, there will be a field “Forgot the password?”, once selection the user is directed to reset the password where the user will be asked for security questions and upon successful authentication of the questions, the user is allowed to reset the password.

**Requirement Specification** - ICDE-Skill Stack System shall collect the required fields: username, answers for security questions, upon successful authentication, can reset the password.

### Sub-Requirements –

- SRS\_0401** ICDE-Skill Stack System shall have forgot password option on login screen.
- SRS\_0402** Forgot password page shall have a security check to authenticate the user.
- SRS\_0403** Upon successful authentication, ICDE-Skill Stack System shall take new password from user.

**SRS\_0404** ICDE-Skill Stack System shall update a user profile with an updated password and store it in Database.

**Tasks -**

**SRT\_0401\_01 Add "Forgot Password" option to login page.**

"Forgot Password" is added in the login page in case if the user forgets the password. / The login page must have Forgot Password option in case if the user forgets the password.

**SRT\_0402\_01 Create a Reset Password page.**

Reset Password page must have security questions fields and New Password fields to collect from the user

**SRT\_0403\_01 Validate answers to security questions.**

The answers given to the security questions are validated.

**SRT\_0403\_02 Collect new password from user.**

The system collects the new password from the user and updates it with the old one.

**SRT\_0403\_03 Password hashing in forgot password page.**

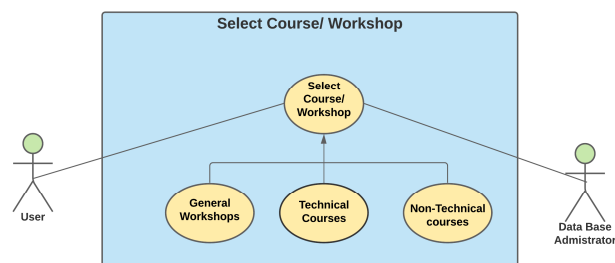
The password entered is hashed and stored for the purpose of security.

**SRT\_0404\_01 Update the user Profile**

The user profile is then updated with the new set of information given by the user.

*Course/Workshop selection*

**Use case - Course/Workshop selection**



*Fig 2.5 Course/Workshop selection use case diagram*

**User Story-** User who wish to select workshop/ course, once he lands on the mainpage there will be three fields – “General workshops”, “Technical Courses” & “Non-Technical Courses”. If the user selects “General workshops”, he’ll then be the given various options to choose according to his convenience. If the user selects for “Technical”, “Non-Technical”, Courses, the user is given various options under different titles to choose from.

**Requirement Specification -** Once after successful login, ICDE-Skill Stack System shall have a main page displaying with General Workshop, Technical course and Non-Technical course sections so that user can select a course from the list of courses inside these course sections conveniently.

**Sub-Requirements –**

**SRS\_0501** ICDE-Skill Stack System shall have main page displaying General Workshop, Technical course and Non-Technical course sections.

**SRS\_0502** ICDE-Skill Stack System Shall allow user to select courses under General Workshop, Technical course and Non-Technical course sections.

**Tasks-**

**SRT\_0501\_01 Create application main page.**

Create application main page with General Workshop, Technical course and Non-Technical course sections display.

**SRT\_0502\_01 Create course section page.**

Create a page that displays list of related courses under that section.

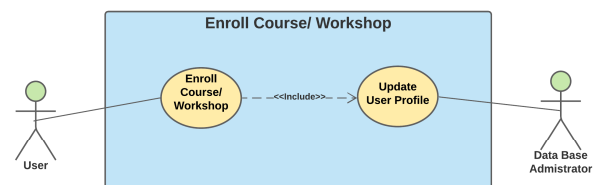
- Create General Workshop course section page.
- Create Technical course section page.
- Create Non-Technical course section page.

**SRT\_0502\_02 Navigate to enroll/drop page.**

The user is navigated to a page where he/she can enroll/drop the courses upon the selection of the course.

*Enrollment*

**Use case - Enrollment**



*Fig 2.6 Enrollment use case diagram*

**User Story –** After the user decides on the course/workshop, he/she will be able to access the course only after enrolling. Thus, there will be a field “Enroll”. If the user selects “Enroll” he/she is enrolled. The course enrolled is updated to user profile.

**Requirement Specification -** As user selects a course from the list of courses, the application shall have an enroll option for a selected course and then update the enrolled course onto user profile.

**Sub-Requirements –**

**SRS\_0601** ICDE-Skill Stack System shall take the response from user to enroll for a particular course.

**SRS\_0602** ICDE-Skill Stack System shall update the enrolled course onto user profile.

**Tasks -**

**SRT\_0601\_01 Create Enroll/Drop page.**

The enroll/drop page is created for the purpose of modifying the selection of the course by the user.

#### SRT\_0601\_02 Create enroll option.

Enroll to course upon clicking on enroll option, if user has not enrolled for selected course before.

#### SRT\_0601\_03 Check for enrollment

Display pop up message "already enrolled to the course" upon clicking on enroll option, if user have enrolled for selected course before.

#### SRT\_0602\_01 Update the user profile in Database.

The user profile is updated in the database in accordance to the selection made by the user.

#### Drop a course/workshop

**Use case** - Drop a course/workshop

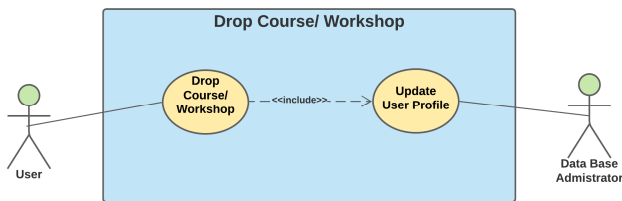


Fig 2.7 Drop a course/workshop use case diagram

**User Story** – Upon enrollment, the user is also given the option of dropping the course at any time. So, if he/she decides to drop the course, there will be a field in every course, “Drop the course”. On selecting it, the user is asked for the reason for dropping it, after reviewing the user is successfully dropped from the particular course.

**Requirement Specification** - For every course/workshop enrolled by the user, application shall have drop the course option.

#### Sub-Requirements --

**SRS\_0701** ICDE-Skill Stack System shall take the response from user to drop a particular course.

**SRS\_0702** ICDE-Skill Stack System shall update the user profile upon dropping a course.

#### Tasks -

##### SRT\_0701\_01 Create Drop option

Drop the course upon clicking on Drop option if user had enrolled for selected course before.

##### SRT\_0701\_02 Check for enrollment

Display pop up message "Not enrolled to the course" upon clicking on Drop option if user had not enrolled for selected course before.

##### SRT\_0702\_01 Update the information in Database

The user profile is updated in the database in accordance to the selection made by the user.

#### Query User Data

**Use case** - Query user data

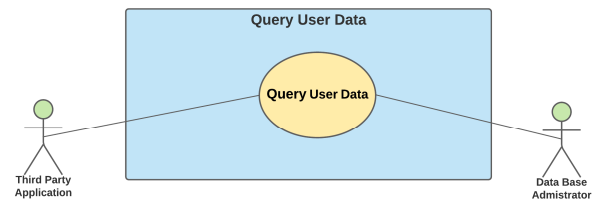


Fig 2.8 Query user data use case diagram

**User Data** - In order to present the statistical report on user actions, the ICDE system queries the data of various users registered in the same discipline and sends it to the 3rd party application.

**Requirement Specification** - As per user's selection, application **shall** query the enrolled workshops and courses list of users of same discipline from the database and passing it to the 3rd party application.

#### Sub-Requirements –

**SRS\_0801** ICDE-Skill Stack System shall query the list of enrolled courses from database sorting user with same discipline.

**SRS\_0802** ICDE-Skill Stack System shall pass the queried data to the 3rd party application

#### Tasks -

##### SRT\_0801\_01 Sorting the course

Sort registered courses among users of same discipline.

##### SRT\_0802\_01 Communicate it with 3<sup>rd</sup> party

Pass sorted data to third party applications.

#### Analyze user pattern

**Use case** – Analyze user pattern

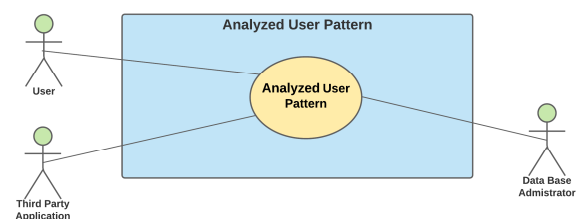


Fig 2.9 Analyze user pattern use case diagram

**User Data** - He/she is in search of a trending course in his/her discipline. In view of achieving it, the administrator who queries the data, analyzes it and presents the data according to certain categories making it easy for the user to select. Those fields include, “Most accessed courses/workshops”.

**Requirement Specification** - Third party Application shall collect the queried data and analyze it to draw a pattern and return the results.

**Sub-Requirements –**

- SRS\_0901** Third party application shall need to analyze the queried data.
- SRS\_0902** ICDE-Skill Stack System shall present the analyzed data to the user.

**Tasks-**

- SRT\_0901\_01 Analyze and store the data**  
Collect the analyzed data from third party application and store it in database.
- SRT\_0902\_01 Recommend courses to users.**  
The course is recommended to the user according to the analyzed data.

**VII. PRODUCT BACKLOG**

Sl. No	Task ID
1.	SRT 0101 01
2.	SRT 0102 01
3.	SRT 0201 01
4.	SRT 0201 02
5.	SRT 0201 03
6.	SRT 0202 01
7.	SRT 0202 02
8.	SRT 0203 01
9.	SRT 0204 01
10.	SRT 0301 01
11.	SRT 0301 02
12.	SRT 0301 03
13.	SRT 0302 01
14.	SRT 0302 02
15.	SRT 0401 01
16.	SRT 0402 01
17.	SRT 0403 01
18.	SRT 0403 02
19.	SRT 0403 03
20.	SRT 0404 01
21.	SRT 0501 01
22.	SRT 0502 01
23.	SRT 0502 02
24.	SRT 0601 01
25.	SRT 0601 02
26.	SRT 0601 03
27.	SRT 0602 01
28.	SRT 0701 01
29.	SRT 0701 02
30.	SRT 0702 01
31.	SRT 0801 01
32.	SRT 0802 01

33.	SRT 0901 01
34.	SRT 0902 01

*Table 2: Product Backlog***VIII. TEAM SETUP & SPRINT CYCLE**

Overall tasks	: 34
Timeline	: 7 Weeks
Duration of 1-Sprint Cycle	: 1 Week
Minimum No of Tasks/Week	: 5

The sprint cycle is followed according to the above given details. Each sprint cycle, an average of six tasks are chosen to complete. The sprint cycle works/functions/operates 5 days a week, which includes work for 2 hours per day to the developer, an additional 30 minutes dedicated for scrum meetings. The changes required to the software design and documentation will be taken up after discussion in scrum meetings and then both product owner and scrum master includes work of 1 hour per day to update these changes and document them.

*Week 1 - 15/02/2021 to 19/02/2021*

**Team member roles:**

Product Owner	: Goutham Kumar Prabhakaran
Scrum-Master	: Ajay Kumar Lakshmipura Vijaykumar
Developer	: Somesh Vemula

**Tasks Taken Up:**

Below mentioned tasks are taken up for week 1 in prioritized order.

Sl. No	Task ID
1	SRT_0101_01
2	SRT_0201_01
3	SRT_0201_03
4	SRT_0202_02
5	SRT_0301_01
6	SRT_0301_02

**Tasks Completed:**

All the tasks mentioned below are completed in prioritized order.

Sl. No	Task ID
1	SRT_0101_01
2	SRT_0201_01
3	SRT_0201_03
4	SRT_0202_02
5	SRT_0301_01
6	SRT_0301_02

*Week 2 - 22/02/2021 to 26/02/2021*

**Team member roles:**

Product Owner	: Somesh Vemula
Scrum-Master	: Goutham Kumar Prabhakaran
Developer	: Ajay Kumar Lakshmipura Vijaykumar

**Tasks Taken Up:**



Below mentioned tasks are taken up for week 2 in prioritized order.

Sl. No	Task ID
1	SRT_0202_02
2	SRT_0302_01
3	SRT_0301_03
4	SRT_0302_02
5	SRT_0401_01
6	SRT_0402_01
7	SRT_0403_01

#### Tasks Completed:

All the tasks mentioned below are completed in prioritized order.

Sl. No	Task ID
1	SRT_0202_02
2	SRT_0302_01
3	SRT_0301_03
4	SRT_0302_02
5	SRT_0401_01
6	SRT_0402_01
7	SRT_0403_01

Week 3 - 01/03/2021 to 05/03/2021

#### Team member roles:

Product Owner : Ajay Kumar LakshmiPura Vijaykumar  
Scrum-Master : Somesh Vemula  
Developer : Goutham Kumar Prabhakaran

#### Tasks Taken Up:

Below mentioned tasks are taken up for week 3 in prioritized order.

Sl. No	Task ID
1	SRT_0403_03
2	SRT_0501_01
3	SRT_0502_02
4	SRT_0502_01
5	SRT_0601_01
6	SRT_0601_02

#### Tasks Completed:

All the tasks mentioned below are completed in prioritized order.

Sl. No	Task ID
1	SRT_0403_03
2	SRT_0501_01
3	SRT_0502_02
4	SRT_0502_01
5	SRT_0601_01
6	SRT_0601_02

#### A. Class Diagram:

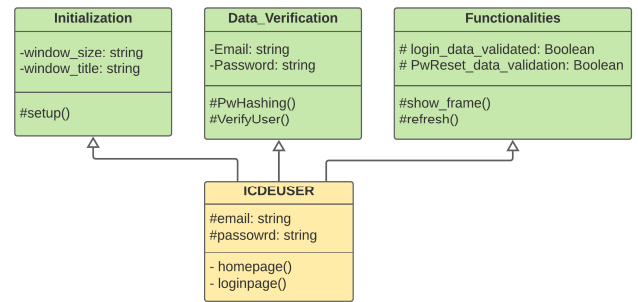


Fig 9.1 Class Diagram for task SRT\_0301\_02

#### B. Sequence Diagram:

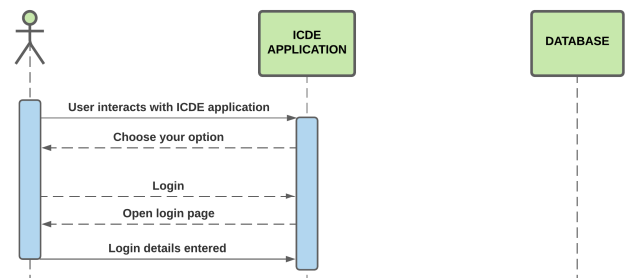


Fig 9.2 Sequence Diagram for task SRT\_0301\_02

#### C. Activity Diagram:

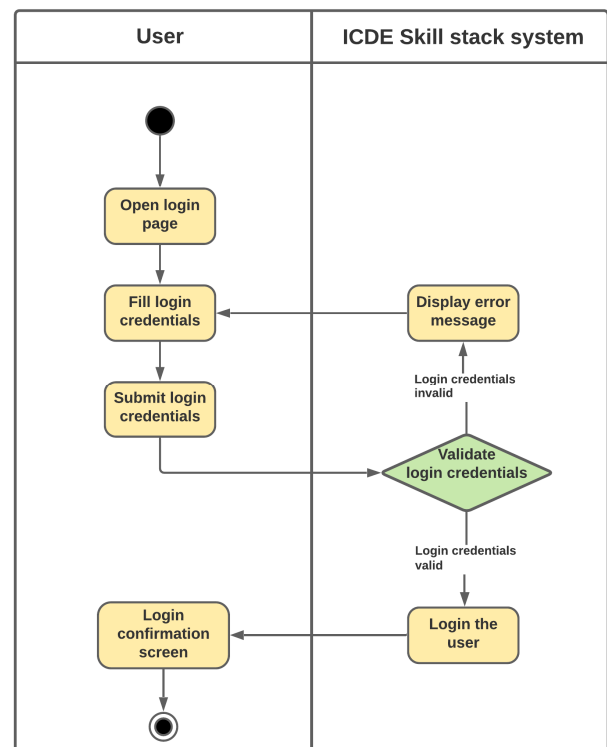


Fig 9.3 Activity Diagram for task SRT\_0301\_02

## IX. MODELLING AND DESIGN OF TASKS

#### Completed task:

SRT\_0301\_02 - Collect user credentials.

The user credentials are collected from the user for logging in.

#### D. Statement Diagram:

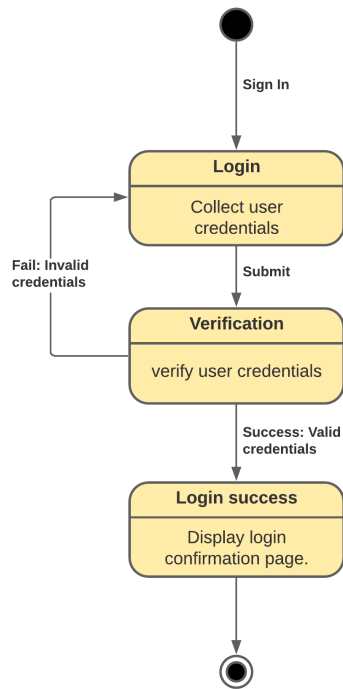


Fig 9.4 Statement Diagram for task SRT\_0301\_02

#### New task:

SRT\_0203\_01 - Create user profile.

The profile is created upon collecting all the information from the user.

#### E. Class Diagram:

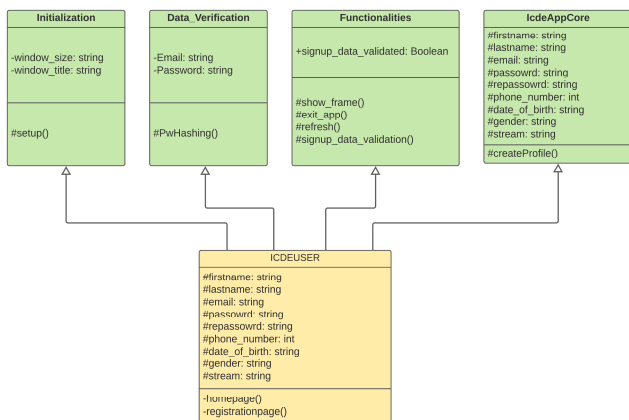


Fig 9.5 Class Diagram for task SRT\_0203\_01

#### F. Sequence Diagram:

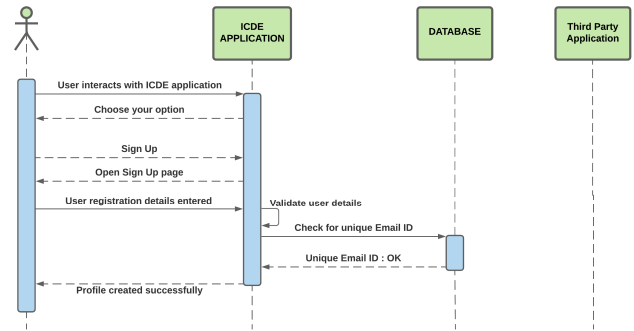


Fig 9.6 Sequence Diagram for task SRT\_0203\_01

#### G. Activity Diagram:

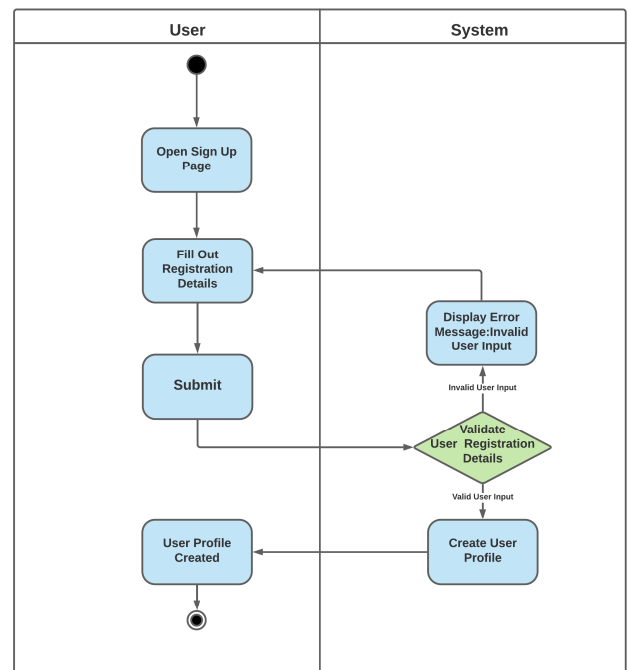


Fig 9.7 Activity Diagram for task SRT\_0203\_01



#### H. Statement Diagram:

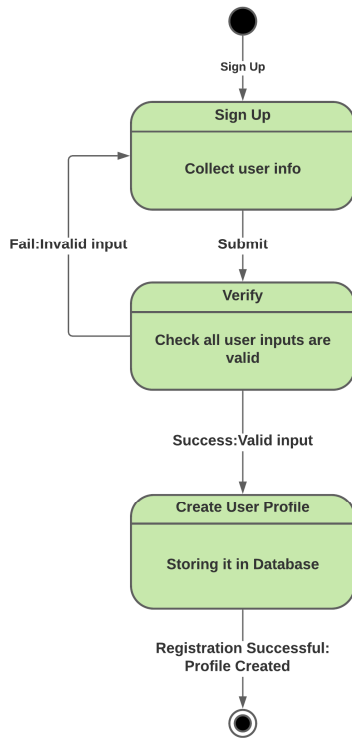


Fig 9.8 Statement Diagram for task SRT\_0203\_01

#### X. EVALUATION

Overall tasks	: 34
Completed tasks	: 19
Remaining tasks	: 15
Remaining Timeline	: 4 Weeks
Duration of 1-Sprint Cycle	: 1 Week
Minimum No of Tasks/Week	: 4

As the number of tasks that needs to be completed in remaining 4 weeks of time duration is 15. Therefore, the minimum number of tasks per week is reduced to 4 tasks/week. Since, the tasks that will be taken up in future might take bit longer duration for the development and documentation of changes to the design. Therefore, the scrum working hours were kept same as before.