

**International School**

**Team projects**

**SRS DOCUMENT**

*Date: November, 2023*

**CMU-SE 214 KIS**

**TO-DO LIST SYSTEM**

**Posted by**

Dao Phuoc Long

Pham Bao Dat

Nguyen Nam

Tran Thanh Hieu

Nguyen Quang Vu

**Approved by**

*Nguyen Trung Thuan*

|  | **PROJECT INFORMATION** | |  |
| --- | --- | --- | --- |
| **Project**  **Abbreviate** | Character's virtual assistant | |  |
| **Project Name** | TO-DO LIST SYSTEM | |  |
| **Start date** | 6/11/2024 | |  |
| **End date:** | 21/12/2024 | |  |
| **Main organizations** | International School, Duy Tan University | |  |
| **Project Advisor** | M.Sc Thuan, Nguyen Trung | |  |
| **Scrum Master** | Dao Phuoc Long | daophuoclong@dtu.edu.vn |  |
| **Team members** | Pham Bao Dat | phambaodat@dtu.edu.vn |  |
| Nguyen Nam | nam.nguyenx191@gmail.com |  |
| Tran Thanh Hieu | tranthanhhieu@dtu.edu.vn |  |
| Nguyen Quang Vu | nguyenquangvu7@dtu.edu.vn |  |

[**1.**](#_heading=h.30j0zll) **Introduce** 4

[**1.1.**](#_heading=h.1fob9te) **Document purpose** 4

[**1.2.**](#_heading=h.3znysh7) **Scope** 4

[**1.3.**](#_heading=h.2et92p0) **Project Proposal** 4

[**1.4.**](#_heading=h.3dy6vkm) **Bibliography** 5

[**2.**](#_heading=h.1t3h5sf) **Overall Description** 5

[**2.1.**](#_heading=h.4d34og8) **System Environment** 5

[**2.2.**](#_heading=h.2s8eyo1) **Functional Requirements Specification** 5

[**2.2.1.**](#_heading=h.ihv636) **Register users Use Case** 5

[**2.2.2.**](#_heading=h.32hioqz) **User Use Case** 6

[**2.2.3.**](#_heading=h.1hmsyys) **Use Case** 7

[**2.2.4.**](#_heading=h.vx1227) **Help ER Diagram** 9

[**2.2.5.**](#_heading=h.4f1mdlm) **System Work Data Flow Diagram** 9

[**2.3.**](#_heading=h.28h4qwu) **Functional Requirements** 10

[**2.4.**](#_heading=h.1mrcu09) **Non-Functional Requirements** 10

[**3.**](#_heading=h.2r0uhxc) **System Design** 11

[**3.1.**](#_heading=h.1664s55) **Description of procedures and function** 11

[**4.**](#_heading=h.2zbgiuw) **Implementation and Testing** 11

[**4.1.**](#_heading=h.1egqt2p) **Introduction** 11

[**4.2.**](#_heading=h.3ygebqi) **Layout** 12

[**4.2.1.**](#_heading=h.2dlolyb) **Login** 12

[**4.2.2.**](#_heading=h.sqyw64) **Register a new account** 12

[**4.3.**](#_heading=h.3cqmetx) **Layouts report** 14

[**4.3.1.**](#_heading=h.1rvwp1q) **Register layout** 14

[**4.3.2.**](#_heading=h.4bvk7pj) **Login layout** 15

1. **Introduce**
   1. **Document purpose**

The purpose of this document is to provide a detailed overview of the ToDo List System, including its intended functionality, scope, and design. It serves as a reference for developers, project managers, and stakeholders involved in the project, ensuring that the system meets the defined requirements and objectives. The document also outlines the technical and user-based requirements, ensuring clarity in system expectations.

* 1. **Scope**
* The scope of the ToDo List System includes the development of a task management platform aimed at three primary user groups: students, workers, and business professionals. The system will allow users to:
* Create, manage, and track personal and team-based tasks.
* Organize tasks by category, priority, and deadline.
* Collaborate within teams (for business users).
* Receive notifications and reports on task progress and productivity. The system will be scalable and accessible across multiple devices (desktop and mobile) to cater to a wide range of user environments.
  1. **Project Proposal**
* The ToDo List System project proposes the creation of a web-based application that offers a user-friendly interface for task management. The system will feature the following key functionalities:
* User registration and role-based access.
* Task creation, modification, categorization, and prioritization.
* Collaboration features for team-based task delegation (business users).
* Customizable views and filters for task management (list view, calendar view).
* Notification and alert system for task deadlines.
* Reporting and analytics for task completion and productivity. The development will follow a multi-tier architecture with a clear separation between frontend, backend, and database layers, ensuring scalability, performance, and security.

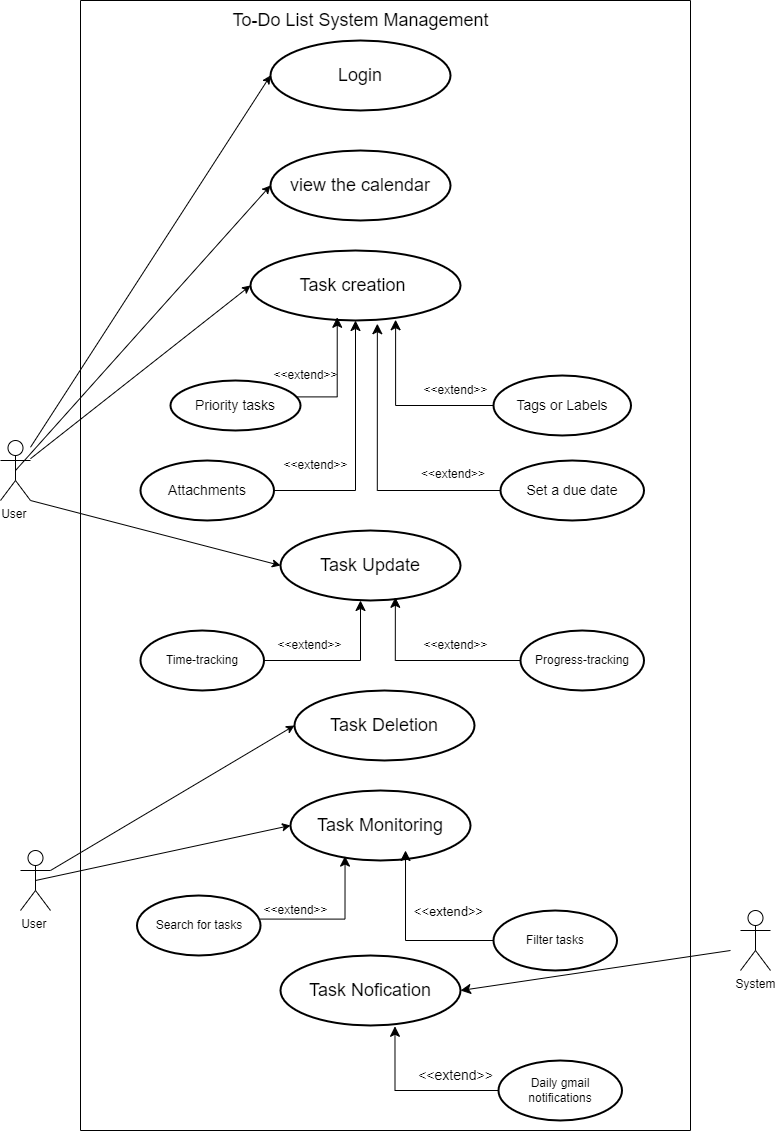
* 1. **Bibliography**

[SRS Example (mu.edu.sa)](https://m.mu.edu.sa/sites/default/files/content/2019/10/Final%20report.pdf)

[Software Requirements Specification document with example - Krazytech](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database)

[Example Software Requirements Specification (SRS) | ReqView](https://www.reqview.com/doc/iso-iec-ieee-29148-srs-example/)

1. **Overall Description**
   1. **System Environment**

****

* 1. **Persona and Scenario**

**2.2.1. Persona**

- Name: Le Huu Tien

- Age: 22

- Job: A third year student majoring in marketing and currently interning at an advertising company.

- Goal: Look for tools that help break down tasks effectively and easily, also the interface easy to see.

- Trouble: don’t know which app to use because the existing app is not easy to use and difficult to see.

- Technology: Laptop, phone, Git,…

**2.2.2. Scenario**

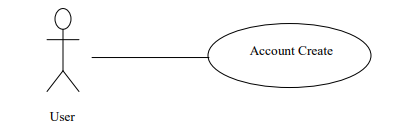
**- Scenario 1:**

* Context: Le Huu Tien need to track his daily work progress and review the total amount of work in last week.
* Action: Le Huu Tien use work management system to create daily tasks and when completed, he will tick the completion box. After a week, the system will summarize and notify him via email or application.
* Result: The work was completed on time, on schedule, helping him increase its income.

-**Scenario 2:**

* Context: Le Huu Tien needs to check important dates quickly without using a calendar.
* Action: Le Huu Tien creates important dates in the management system. When the date arrives, the system will notify via email and right on the phone.
* Result: Le Huu Tien will not forget important dates because the system will notify him twice a day.
  1. **Functional Requirements Specification**
* **Login** 
  + **Register**
    - **Address**
    - **Phone number**
* **Search for tasks**
* **Task Management**
  + **Create Tasks**
    - **Attach file**
    - **Tags or Labels**
  + **Delete Tasks**
  + **Task Classfication**
  + **Change of Tasks**
    - **Time-tracking**
    - **Progress-tracking**
* **Monitor the progress of the tasks**
  + **Monitoring tasks status**
  + **Reporting work**
  + **Task history**
  + **Mark the task**
* **Task Announcement** 
  + **Set up notifications** 
    - **Daily gmail notifications**
    - **Set notification time**
* **Maintaince**
  + 1. **Register users Use Case**

***Use case***: Account Create

****

***Brief Description***

If users want to access the application, they must register to be granted access.

***Initial Step-By-Step Description***

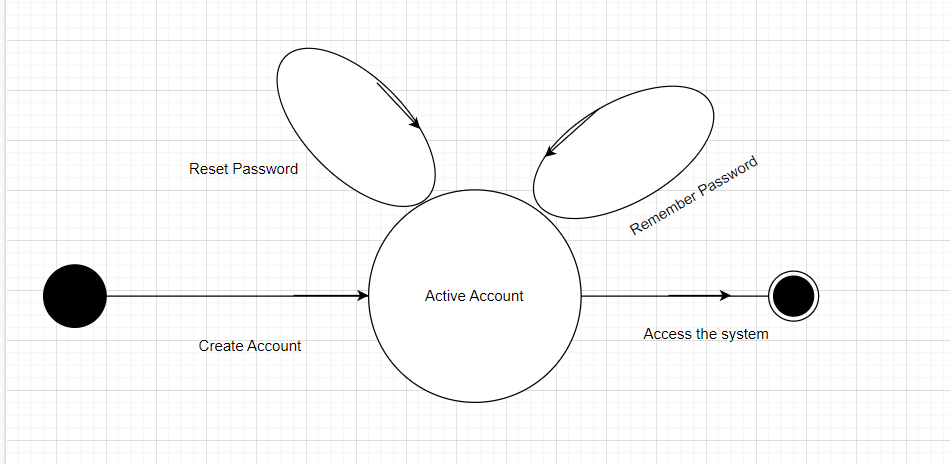
Before this use case can be initiated, the user already has the program.

1. The user will create an account.

2. The system requires the user to register an account.

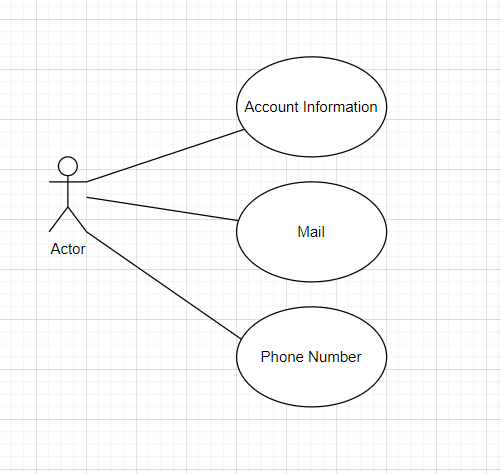
3. The user needs to fill in the information on the form.

4. The user will be granted access to the system by the system.

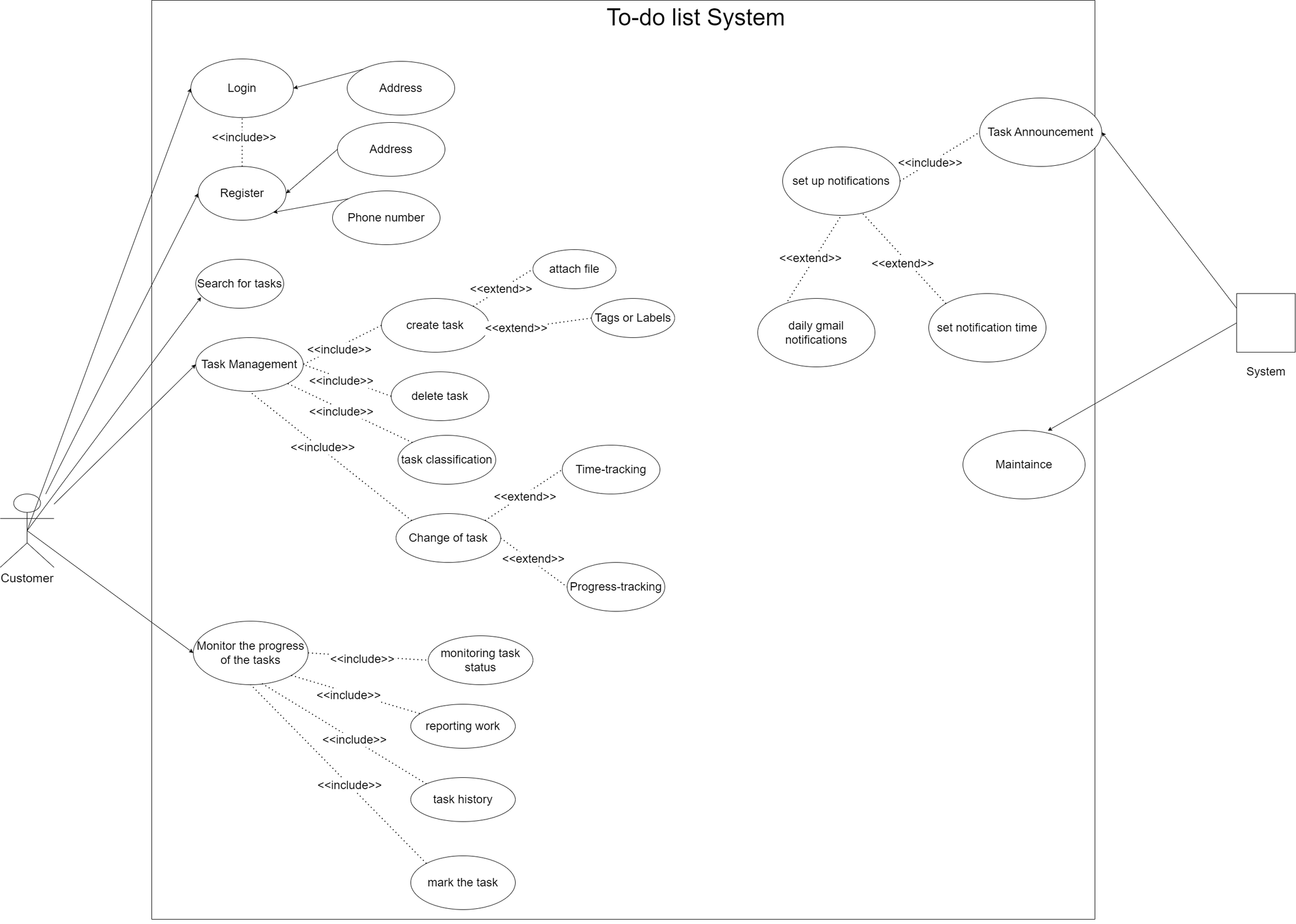
****

* + 1. **User Use Case**

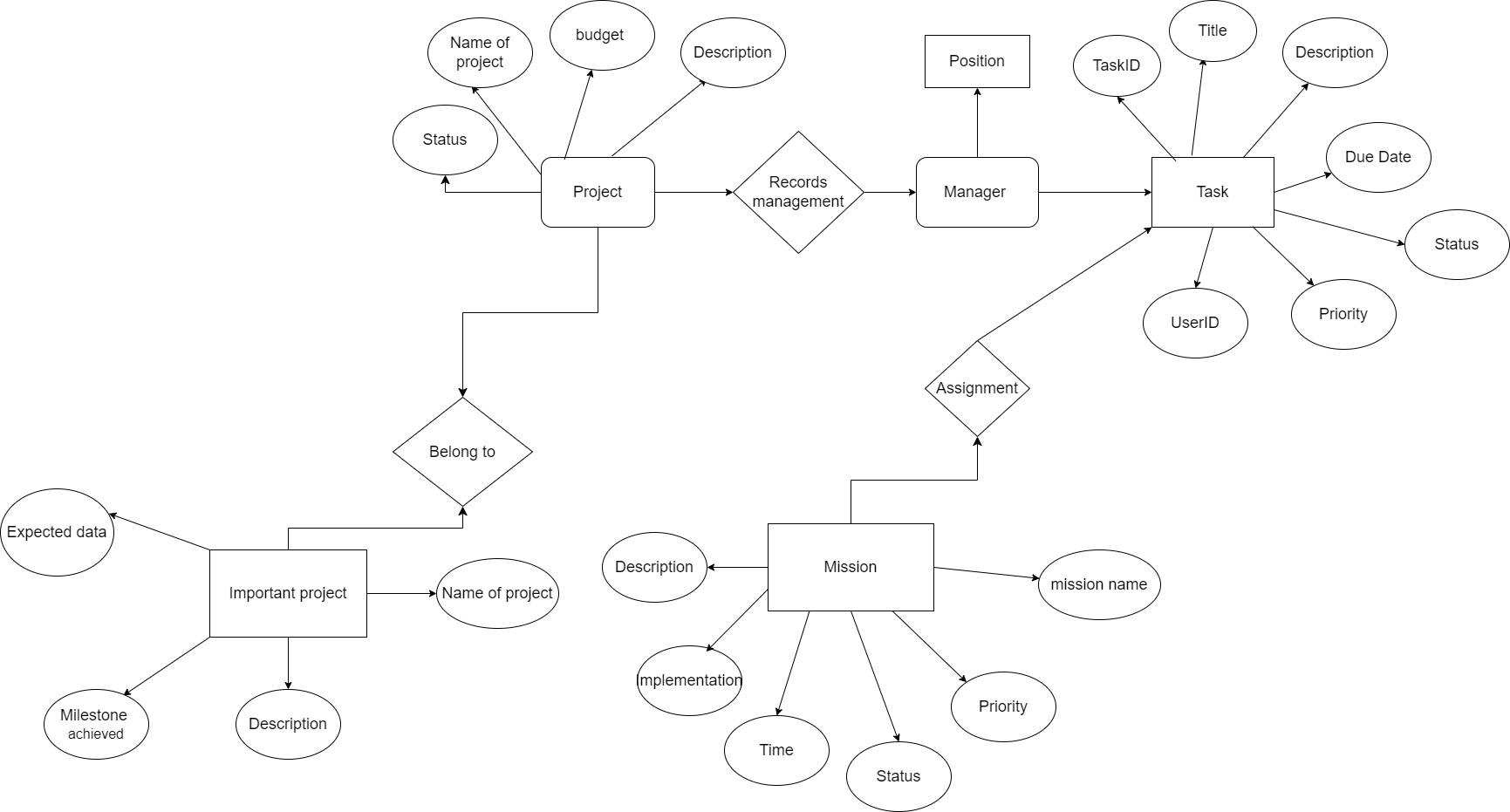
The User after registering has the following:

****

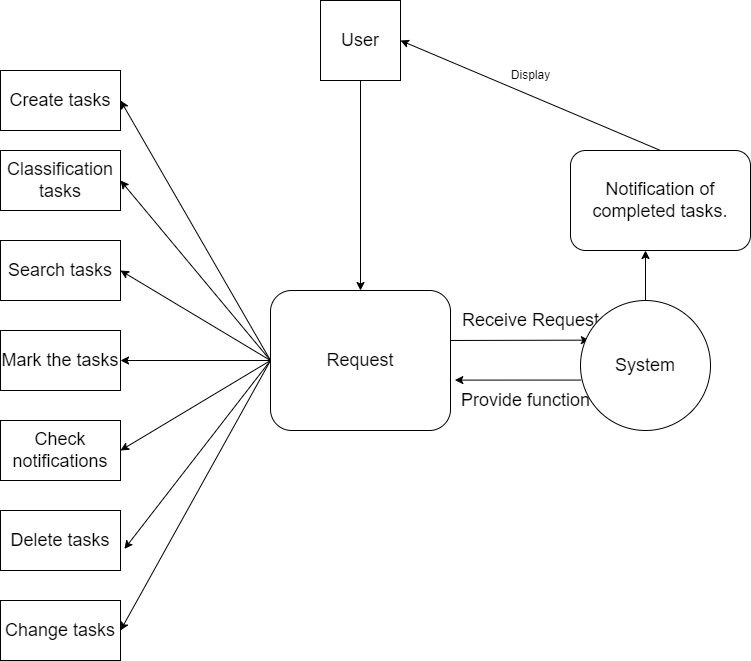
* + 1. **Use Case**

****

* + 1. **Help ER Diagram**

****

* + 1. **System Work Data Flow Diagram**

****

* 1. **Functional Requirements**

**2.4.1 User Management:**

* Users can register and log in.
* Different roles for students, workers, and businesses.
* Users can reset passwords.

**2.4.2 Task Management:**

* Users can create tasks with titles, descriptions, due dates, and priorities.
* Users can categorize tasks (e.g., by subject for students, project for workers, etc.).
* Tasks can be updated or deleted.
* Users can mark tasks as completed.
* Notifications or reminders for upcoming deadlines.

**2.4.3 Task Views and Filters:**

* Users can view tasks in different formats (list view, calendar view).
* Users can filter tasks by status, priority, or category.

**2.4.4 Collaboration (for business users):**

* Users can assign tasks to team members.
* Team members can update task progress.
* Team leaders can monitor overall task completion.

**2.4.5 Analytics:**

* Users can view statistics about task completion.
* Productivity reports showing completed tasks, time taken, etc.

* 1. **Non-Functional Requirements**

**2.5.1 Performance:**

* The system should be able to handle multiple users simultaneously without delay.
* Response time should be under 3 seconds for loading tasks or reports.

**2.5.2 Security:**

* User data must be encrypted, especially passwords.
* Role-based access control (e.g., admin, user) for sensitive operations.
* Regular system audits to check for security vulnerabilities.

**2.5.4 Usability:**

* The system should be intuitive, with an easy-to-navigate interface.
* The system should be accessible on both desktop and mobile platforms.

**2.5.5 Scalability:**

- The system must support a growing number of users without performance degradation.

**2.5.6 Availability:**

* The system should have 99.9% uptime to ensure reliability.
* Automated backups to prevent data loss.

**3. System Design:**

### 3.1 User Flow Design:

* Create a **User Journey Map** to visualize the steps a user (student, worker, business professional) will take from registration to task management.
* Highlight key interactions such as login, task creation, task update, and notifications.
* Include branching paths for different user roles (students might organize tasks by subjects, workers by projects, businesses by teams).

### 3.2 Wireframes:

* Develop low-fidelity wireframes to establish the layout and structure of each screen.
* **Login & Registration Page**: Simple, clean forms to sign in or sign up, with fields for role selection (student, worker, business).
* **Dashboard**: A central hub where users can view all their tasks. Separate sections for:

l Upcoming tasks

l Completed tasks

l Filters (by priority, deadline, etc.)

* **Task Creation Form**: A form to input task title, description, due date, priority, and category.
* **Task Details Page**: Displays detailed information for each task, with options to edit, delete, or mark as complete.
* **Calendar View**: An alternative view of tasks based on deadlines.
* **Team Collaboration (for business users)**: A section to delegate tasks to team members and track their progress.

### 3.3 High-Fidelity Mockups:

* Add visual design elements (colors, fonts, icons) that match the brand identity and usability standards.
* **Color Scheme**: Choose colors that signify priority (e.g., red for high priority, green for completed tasks).
* **Typography**: Select a readable, modern font (e.g., Roboto, Open Sans).
* **Buttons & Icons**: Use clear call-to-action buttons like "Create Task," "Complete Task," and task-related icons (calendar, checkboxes).
* Mockups should ensure **responsiveness**, adjusting the layout for different devices (desktop, tablet, mobile).

### 3.4 Prototyping:

* Link the screens together using Figma's prototyping tools to simulate the interaction flow.
* Clicking on a task should take the user to the Task Details page.
* Task filters should dynamically show relevant tasks based on user preferences. Include animations/transitions for smooth UX.

### 3.5 Component Library:

* Create a **Design System** within Figma for reusable components like buttons, form fields, modals, task cards, and filters.
* This will standardize the UI elements across different screens and make development faster.

### 3.6 Collaboration Tools:

* Utilize Figma's collaboration feature to allow real-time feedback and co-designing with team members.
* Stakeholders can leave comments on specific elements, and developers can extract the necessary design specifications.

1. **Implementation and Testing**
   1. **Introduction**

This application is implemented in Figma.

It contains three basic interfaces:

1. Register a new account

2. Log in

3. Main page

Three dialog boxes:

1. Home screen

2. Add new users

3. Add comments

And three databases:

1. User database

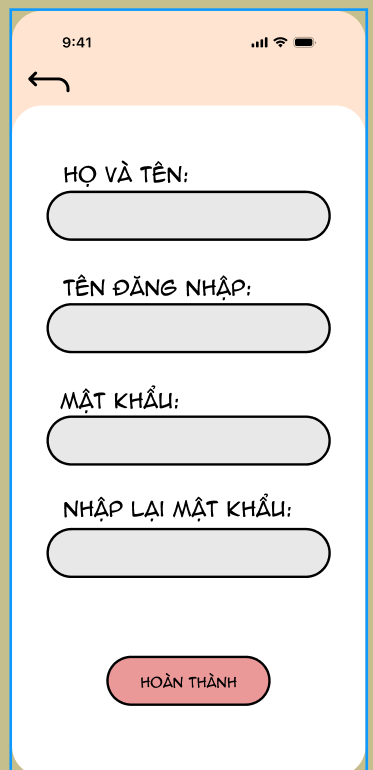
2. Information database

3. Comments database

* 1. **Layout**
     1. **Log in**

****

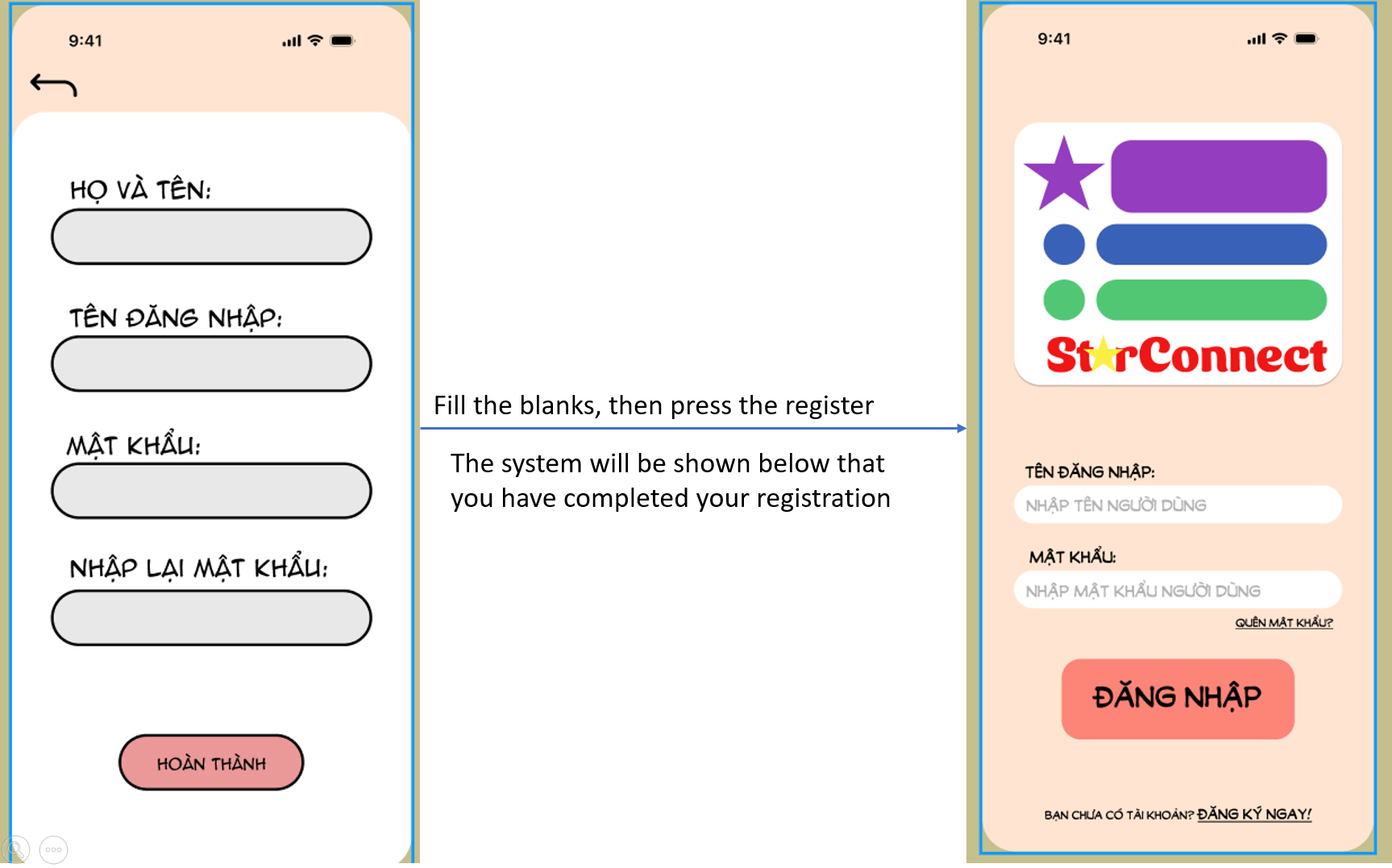
* + 1. **Register a new account**

****

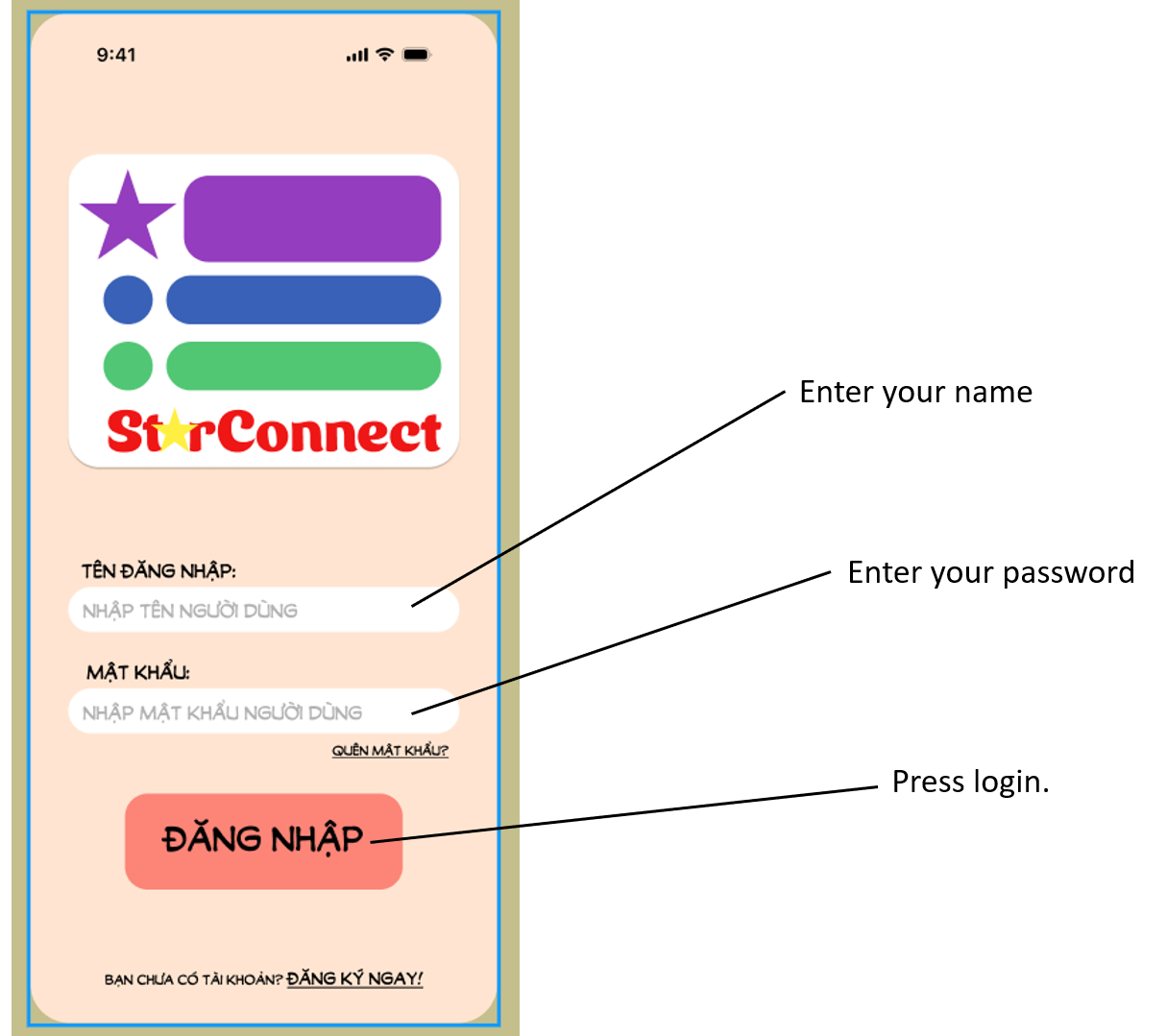
* + 1. **Home**

****

* 1. **Layouts report** 
     1. **Register layout**

****

* + 1. **Login layout**

****