

**International School**

**Requirements Engineering Project**

CMU-SE 214

**Project Proposal**

**Version 1.0**

**Date: September 27, 2024**

**StarConnect – ToDo List System for student, worker, business**

**Created by Group 1**

**Pham Bao Dat**

**Dao Phuoc Long**

**Tran Thanh Hieu**

**Nguyen Nam**

**Nguyen Quang Vu**

**Approval of Mentor:**

Name Signature Date

#### **PROJECT INFORMATION**

| **Project acronym** | **StarConnect** | | |
| --- | --- | --- | --- |
| **Project Title** | The ToDo List System that help everyone manage their works | | |
| **Start Date** | 27 Sep 2024 | **End Date** | 11 Oct 2024 |
| **Lead Institution** | International School, Duy Tan University | | |
| **Project Mentor** | Thuan, Nguyen Trung, Ph.D | | |
| **Scrum master**  **/ Project Leader & contact details** | Long, Dao Phuoc  Ema[il: da](mailto:thanhsang3111999.ntb@gmail.com)ophuoclong@dtu.edu.vn  Tel : 0563730648  ID: 29219054593 | | |
| **Partner Organization** |  | | |
| **Project Web URL** |  | | |
| **Team members** | Name | Email | Tel |
| 29219054593 | Long, Dao Phuoc | daophuoclong@dtu.edu.vn | 0563730648 |
| 29216250704 | Dat, Pham Bao | phambaodat@dtu.edu.vn | 0794710341 |
| 29219034776 | Nam,  Nguyen | nam.nguyenx191@gmail.com | 0364083266 |
| 29219065013 | Hieu,  Tran Thanh | tranthanhhieu4@dtu.edu.vn | 0935092531 |
| 28211352230 | Vu,  Nguyen Quang | nguyenquangvu7@dtu.edu.vn | 0795644886 |

**REVISION HISTORY**

| **Version** | **Date** | **Comments** | **Author** | **Approval** |
| --- | --- | --- | --- | --- |
| 1.0 | 20/02/2021 | Initial Release | All members |  |

**1. Project Title**

"**To Do List System for student, worker, business**"

**2. Project Overview**

* **Main goal:**This project aims to create an effective task management system that helps users organize and track daily tasks. Main audiences include students, working people and business.
* **Short description:** Helps users create to do lists,set deadlines,reminders,and classify tasks by priority. User friendly interface for many user groups

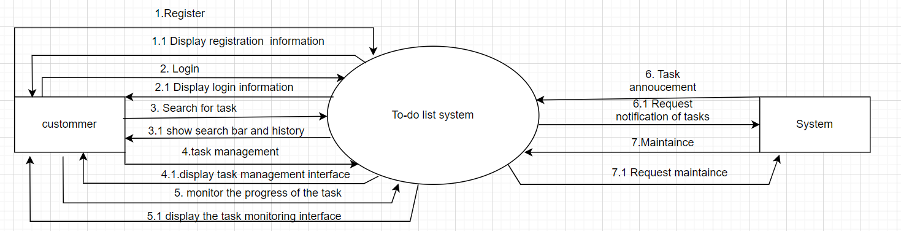
**3. Project Background and Motivation**

* **Background and Motivation:**In today’s fast-paced word, people often juggle multiple responsibilities across different roles students managing academic schedules ,workers balancing job tasks, and businesses coordinating team activities. With increasing demands on time ,there is a growing need for and affective tool to manage tasks ,deadlines , and priorities

**4. Proposed Solution**

* **The proposed solution:** help students, workers, and businesses efficiently organize and prioritize their tasks. This system will utilize natural language processing (NLP) to understand user inputs in a conversational manner. It will enable users to add, modify, or delete tasks through simple, intuitive commands.The system is designed to be simple and effective,The system uses fig ma to design UI and UX and using Python as the core programming language

* **Context diagram** :

**

**5. Related Works or Projects on the Market**

* Some applications on the market, such as Todoist, Trello, and Any.do, require payment to unlock all of their features. Our application, however, not only provides a full set of essential functionalities but is also completely free and easy to use, even for beginners.

**Create a table for a comparison of these products.**

| **Characteristics** | **Todoist** | **Trello** | **Microsoft**  **To do** | **Any.do** | **To do list(New)** |
| --- | --- | --- | --- | --- | --- |
| **Create and manage tasks** | **+** | **+** | **+** | **+** | **+** |
| **Task classification** | **+** | **+** | **+** | **+** | **+** |
| **Task reminder** | **+** | **+** | **+** | **+** | **+** |
| **Integrated with calendar** | **+** | **+** | **+** | **+** | **+** |
| **Real reviews**  **from customers** | **+** | **+** | **+** | **+** |  |
| **Multi languages support** | **+** | **+** | **+** | **+** |  |
| **Simple interface, easy to use** | **+** | **+** | **+** | **+** | **+** |
| **Monitor the progress of the tasks** | **+** | **+** | **+** | **+** | **+** |
| **Free to use** |  |  | **+** | **+** | **+** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**6. Objectives and Deliverables**

* **Objective**
* **Objective 1**: Develop a fully functional to do list interface.
* **Deliverable**: An application with full functions, easy to use, helping people complete tasks

**7. Methodology and Tools**

**Kanban** is a work management method that helps visualize workflows, focusing on optimizing processes and ensuring a continuous flow of tasks.

- **Create a Kanban Board**

The **Kanban Board** is the most essential tool in this method. It is usually divided into columns, with each column representing a stage in the workflow process.

Common columns typically include:

**+ Backlog**: Where all tasks that haven't started yet are stored.

**+ In Progress**: Tasks currently being worked on.

**+ Review/Testing**: Tasks undergoing review or testing.

**+ Done**: Tasks that have been completed.

Each task is represented by a card that moves across the columns as the work progresses.

- **Define and Limit Work in Progress (WIP)**

Set WIP limits for each column to control the number of tasks that can be in progress at the same time.

WIP limits help prevent work overload and ensure the quality of tasks is not compromised.

- **Prioritize and Categorize Work**

Organize the tasks in the **Backlog** column by priority, making it easier for the team to select what to work on next.

Tasks can be categorized based on importance or urgency.

- **Pull System for Continuous Flow**

Instead of pushing tasks from managers to the team, team members proactively "pull" tasks from the **Backlog** or previous columns when they have the capacity to handle them.

This helps avoid overloading and ensures an optimized workflow.

- **Continuous Improvement**

Use **Kaizen** sessions or short meetings to evaluate the workflow and find ways to improve it.

Analyze "bottlenecks" in the workflow and develop solutions to eliminate or reduce delays.

- **Track and Measure Performance**

**+ Lead Time**: The time from when a task is started until it is completed.

**+ Cycle Time**: The time from when a task is pulled into the **In Progress** column until it is finished.

These metrics help the team evaluate performance and find ways to optimize the workflow.

- **Continuously Improve the Workflow**

Regularly review and optimize the workflow to improve task speed and quality.

- **Retrospective meetings** can help assess the process and gather feedback from team members for continuous improvement.

· **Tools and Technologies**

- **Backend:**

**+ Python** with the **Flask** framework for developing APIs and server-side logic.

**+ TensorFlow** for AI/ML functionalities such as emotion detection or natural language processing.

- **Frontend:**

**+ React** to build flexible and dynamic user interfaces.

**+ HTML5/CSS3** for static interface design and styling, ensuring mobile-friendly responsiveness.

- **Database:**

**+ PostgreSQL** for managing and storing relational data.

**+ Redis** for caching and improving response times for repeated requests.

- **Version Control:**

**+ Git** for version control, with repositories hosted on **GitHub** for team collaboration and code reviews.

- **CI/CD:**

**+ Jenkins** for continuous integration and automated deployment, ensuring the code is regularly tested and deployed without much manual intervention.

- **Cloud Platform:**

**+ AWS** or **Google Cloud** for scalable infrastructure to host backend services, databases, and storage.

- **Project Management:**

**+ Jira** or **Trello** to manage tasks within sprints, track progress, and maintain transparency among team members.

**8. Timeline**

| **Phase** | **Task** | **Duration** | **Estimated Completion** |
| --- | --- | --- | --- |
| Week 1-2  **Project research and requirements garthering** | + Reseacrch technologies and tools  + Define project requirements  + Set up project requirement tools | 2 weeks | End of week 2 |
| **Weeks 3-4**  **Backed development ( API & logic)** | + Set up flask famewwork  Develop core backend APIs  + Integrate Postgre API & Redis | 2 weeks | End of week 4 |
| Weeks 5-6  **AI/ML integration (TensorFlow)** | + Implement emotion detection or NLP  + Integrate AL/NL models into backend | 2 weeks | End of week 6 |
| Weeks 7-8  **Frontend development** | + Set up React framework  + Build core user interface  + Implement API connections | 1 weeks | End of week 8 |
| Week 9  **Testing and debugging** | + Perform unit and integration testing  + Fix bugs and optimize performmance | 1 weeks | End of week 9 |
| Weeks 10  **Final deployment and presentation** | + Deploy to AWS or Google Cloud  + Final review and presentation | 1 weeks | End of week 10 |

**Chart Gantt**: [https://app.clickup.com/9018556022/v/li/901803518725](https://app.clickup.com/9018556022/v/g/8crrfkp-418)

**Note:** Gantt charts are valuable tools in project management that help ensure tasks are completed on time and provide an overview of the project's status. Using Gantt charts can enhance team productivity and improve coordination among team members.

**9. Project Team**

| **Team member** | **Role** | **Responsibilities** |
| --- | --- | --- |
| Long | Project manager | Oversees project progress, manages the timeline, and ensures team communication. |
| Dat | Leader development | Responsible for backend development, API creation, and chatbot integration. |
| Nam | Frontend Development | Develops the user interface using React, ensuring responsiveness and user experience. |
| Hieu | AL/ML Engineer | Implements AI/ML models for emotion detection and natural language processing (TensorFlow). |
| Vu | DevOps Engineer | Manages cloud infrastructure (AWS/Google Cloud), CI/CD pipeline, and deployment processes. |
| A | Quality Assurance Lead | Conducts testing, debugging, and ensures product quality through unit and integration tests. |
| B | UI/UX Designer | Designs the user interface layout and ensures a smooth user experience across all devices. |

**10. Risk Management**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Delayed Integration of AI/ML Models | Schedule buffer time in the timeline, begin integration testing early, and prioritize model integration. |
| Technical Challenges with API and Database Integration | Conduct thorough research during the initial phases and perform regular unit testing to catch early issues. |
| Incomplete Requirements Gathering | Organize regular check-ins with stakeholders and conduct early requirement reviews to prevent scope creep. |
| Overload of Work in Progress (WIP) During Development | Limit WIP using Kanban methodology, and ensure team members pull tasks only when they have capacity. |
| Cloud Infrastructure Cost Overruns | Regularly monitor cloud usage, set budget alerts on AWS/Google Cloud, and optimize resource scaling. |
| Poor Performance or Bugs in Production | Implement a thorough testing strategy, including unit, integration, and load testing before deployment. |
| Lack of Communication Between Team Members | Schedule daily stand-ups or regular check-ins to keep communication flowing and address issues early. |
| Unforeseen Delays in Frontend-Backend Integration | Establish clear API contracts between the frontend and backend teams, and perform early integration tests. |
| Difficulty in Maintaining CI/CD Pipeline | Regularly monitor Jenkins pipelines, automate testing, and allocate resources to address issues promptly. |
| Resource Constraints or Team Unavailability | Plan for backups in case of team unavailability and cross-train team members to handle multiple tasks. |

**11. Budget and Resources**

| **Category** | **Item** | **Estimated Cost** |
| --- | --- | --- |
| Software Licenses | TensorFlow (Open-source)  Flask (Open-source)  PostgreSQL (Open-source)  Redis (Open-source)  React (Open-source)  API Access (NLP/Emotion Model) | $0  $0  $0  $0  $0  $0  $50/month |
| Hosting and Infrastructure | AWS or Google Cloud (Server hosting) | $200/month |
| Hardware | Development laptops (for team)  Backup hardware (external drives) | Provided by company    $150 |
| Human Resources | Developers (Salaries)  AI/ML Engineers (Salaries)  DevOps Engineer (Salaries)  QA Engineers (Salaries) | Based on project scope  Based on project scope  Based on project scope  Based on project scope |
| Miscellaneous | Jenkins CI/CD setup | $50 |

**Total Estimated Monthly Budget: $450**

**Resources**

**- Human Resources:**

+ 2 Backend Developers

+ 1 Frontend Developer

+ 1 AI/ML Engineer

+ 1 DevOps Engineer

+ 1 QA Engineer

+ Project Manager

**- Hardware:**

+ Development laptops for the entire team

+ Backup storage solutions (external drives or cloud backup)

**- Software:**

+ Flask (Backend framework)

+ TensorFlow (AI/ML model)

+ PostgreSQL (Database)

+ Redis (Caching)

+ React (Frontend development)

**- Hosting:**

+ AWS or Google Cloud for server hosting

+ Cloud storage and compute resources

**12. Project constraints**

| **Constraint** | **Constraints Description** | **Guidelines for Acceptance** |
| --- | --- | --- |
| **Economic** | Ensure the project remains within budget, and evaluate the total cost of ownership throughout the product lifecycle. | Elements for consideration are design costs, production costs, maintenance costs, operating costs, and sales price |
| **Environmental** | Conduct an environmental impact assessment and prioritize sustainable materials and practices in the design process. | The impact of the design on the environment as well as the impact of the environment (e.g. temperature range, humidity, vibration, electromagnetic interference immunity, and shock) on the design should be considered. Design for recycling and design to use recycled materials should also be considered |
| **Ethical** | Ensure compliance with ethical standards and regulations, and conduct a thorough review of potential ethical dilemmas. | Ethical considerations can be broad. Areas that are typically addressed include intellectual property, reverse-engineering, privacy, security, and the conflict between cost and safety |
| **Public health, safety, and welfare** | Adhere to industry safety standards and conduct user testing to ensure the product does not pose health or safety risks. | Includes safety standards as well as the impact of the design on users (for example, electrical or physical hazards) |
| **Social and Global** | Engage with stakeholders to understand societal needs and ensure the design meets global and socially responsible engineering standards. | Addresses aspects such as benefits, risks, the man-machine interface, the acceptance of products by the intended user or by society at large, and global and socially responsible engineering. |
| **Cultural** | Conduct cultural assessments to understand diverse user needs and preferences, ensuring designs are culturally sensitive. | Which cultural characteristics could influence the approach?  How do the design from different cultures differ? |
| **Sustainability** | Implement sustainable practices in sourcing materials and design to ensure durability, reusability, and maintainability. | Refers to the sustainability of resources, including material, energy, supplies, manufacturing techniques, personnel, operation, and the need for additional infrastructure, as well as the sustainability of the design including reliability, lifetime, durability, reusability, maintainability. |

**13. Conclusion**

This project represents a significant advancement in the integration of AI and emotional intelligence within software applications, particularly in the realm of user support systems. By developing an intelligent chatbot that leverages natural language processing (NLP) and machine learning, we aim to provide users with immediate and personalized assistance. This innovative solution is designed to enhance user engagement and satisfaction by offering tailored responses that address individual needs.

The anticipated impact of this project extends beyond just improving user experience; it sets a new standard for how software can interact empathetically with users. As the demand for responsive and intuitive applications continues to grow, our project will contribute valuable insights and methodologies to the field of Software Engineering. By demonstrating the effective application of AI in enhancing user support, we aim to inspire further research and development in this area, potentially influencing future software solutions across various industries.

In summary, this project not only addresses the immediate needs of users for reliable and accessible support but also fosters advancements in the broader landscape of software development, highlighting the importance of integrating emotional intelligence into technology.

**14. References**

| **No.** | **References** | **Document Information** |
| --- | --- | --- |
| 1 | Scrum Model | <https://en.wikipedia.org/wiki/Scrum_(software_development)> |
| <https://www.atlassian.com/agile/scrum> |
| <https://www.digite.com/agile/scrum-methodology/> |
| <https://www.scrum.org/resources/scrum-guide> |
| 2 | Technical | <https://www.tensorflow.org/tutorials/quickstart/beginner?hl=vi> |
| [https://www.flutterclutter.dev/flutter/tutorials/implementing-edge-detection-in-flutter/2020/1509/](https://www.flutterclutter.dev/flutter/tutorials/implementing-edge-detection-in-flutter/2020/1509/%20) |
| <https://stackoverflow.com/questions/14248571/finding-properties-of-sloppy-hand-drawn-rectangles> |
| <https://www.tutorialspoint.com/how-to-detect-a-rectangle-and-square-in-an-image-using-opencv-python> |
| 3 | Standard | [https://www.nws.noaa.gov/oh/hrl/developers\_docs/General\_So](https://www.nws.noaa.gov/oh/hrl/developers_docs/General_Software_Standards.pdf) [ftware\_Standards.pdf](https://www.nws.noaa.gov/oh/hrl/developers_docs/General_Software_Standards.pdf) |
| <https://standards.ieee.org/standard/12208-2017.html> |
| <https://en.wikipedia.org/wiki/Scrum_(software_development)> |

#### 15. Attached “DESCRIPTION OF PRODUCT REQUIREMENTS”