School of Information, Computer and Communication
Technology Sirindhorn International Institute of Technology
Thammasat University

DES424 Cloud-based Application Development

Kickoff Report

Group 11

Members

Nachat Kaewmeesang	6422770774
Pusit Eiamsook	6422772226
Kittisak Wangananansukdee	6422781441
Krittidech Paijitjinda	6422781664
Supakorn Vannathong	6422782712
Arkaravit Raksakaeo	6722800099

Present to
Dr. Apichon Witayangkurn

Semester 1 Academic Year 2024

Table of Contents

Introduction

This document is a kickoff report for DES424 Cloud-based Application Development project, submitted to Dr. Apichon Witayangkurn. The objective of this report is to describe our ideas and objectives for our project called “TUDU”, an online task management web app.

Problem Statement

Most to-do apps and collaborative task management apps are packed with features and tools to help individuals and teams perform better. Usually, these apps are designed for professional use, such as for developers, designers, architects, and others. The additional features are beneficial for many use cases, but are ill-suited for casual users for most projects. The additional features can be tricky to learn and apply in more casual teams.

TUDU intends to be a lightweight web application that is easy to use for most people, providing just enough functionality to be useful for general cases.

Background

TUDU is a lightweight to-do list app designed for general usage. Whether it be group works among students, or club activities for the Botany Club, TUDU is for them. The app is designed to be easy to learn and even easier to use.

TUDU organises tasks into “cards”. A card can have descriptions, deadlines, tag other users, or other relevant information. Every card is organised into “decks”. A deck contains access lists – view or edit – for users. The creator of a deck is the “owner” who can add other users to view or edit. A deck allows users to track timeline, notifications, and reminders. Additionally, a deck also has a Tableau to visualise progress summary or statistics.

Key Users

The app is intended for casual users, aged 12 to 60 of all various teams and goals. Here are some user persona for our app:

Culture	Casual and friendly – for use in non-professional work
Communication	Commonly use mobile apps, such as LINE or Facebook. As such, the UI of TUDU would be usable on mobile and asynchronous communication.
Use case	For use in non-professional work: clubs, family businesses, hobbies, or other casual works that don't necessitate for a feature rich app.

Requirements

Functional Requirements

Being an online task management app, the users must be able to perform basic tasks regarding user accounts, decks, and cards. All these actions are supported by a dedicated frontend UI and backend’s REST API. Note that these are just still a concept.

Scope	Description
Account	Create account, log-in, modify account, delete account.
Deck	View, create, delete, modify; depending on user’s access rights.
Deck Members	Assign/remove users from deck, selecting view or edit permissions.
Card	View, create, delete, modify; depending on user’s access rights.
Tracking	Timeline, notification, reminder.
Dashboard	Tableau – summary, statistics, or other relevant analysis.
Feedback	Report a bug issue and suggestion for improvement

Non-functional Requirements

The online task management app should provide a smooth and efficient user experience by focusing on system performance, reliability, and usability. These are still concepts.

Scope	**Description**
Accessibility	The frontend should follow a11y accessibility standard.
Usability	Usable and comfortable UI for both desktop and mobile. The UI should adjust positioning accordingly to be usable for all cases.
Response	The app should be real-time or near real-time. The app should be reactive to changes within acceptable limits (a few seconds).
Audit Log	Track user activity
Security	Using Google account for easy and quick sign-ins
Cross-Platform Compatibility	Allow the app to work across different operating systems

Technology

Language	Python (Backend System), HTML/CSS (Web page and decoration), TypeScript (Script for the web page), SQL (Database Language)
Cloud Components	Using AWS – EC2 (web hosting) & RDS (database)
Framework	Svelte (Frontend framework for web page creation), FastAPI (Backend framework for handling HTTPS requests)
Tool	Git (Version control for the codes), Bitbucket (Git platform for version control), Jira (Task Management), Tableau (GUI-based Dashboard Creator)

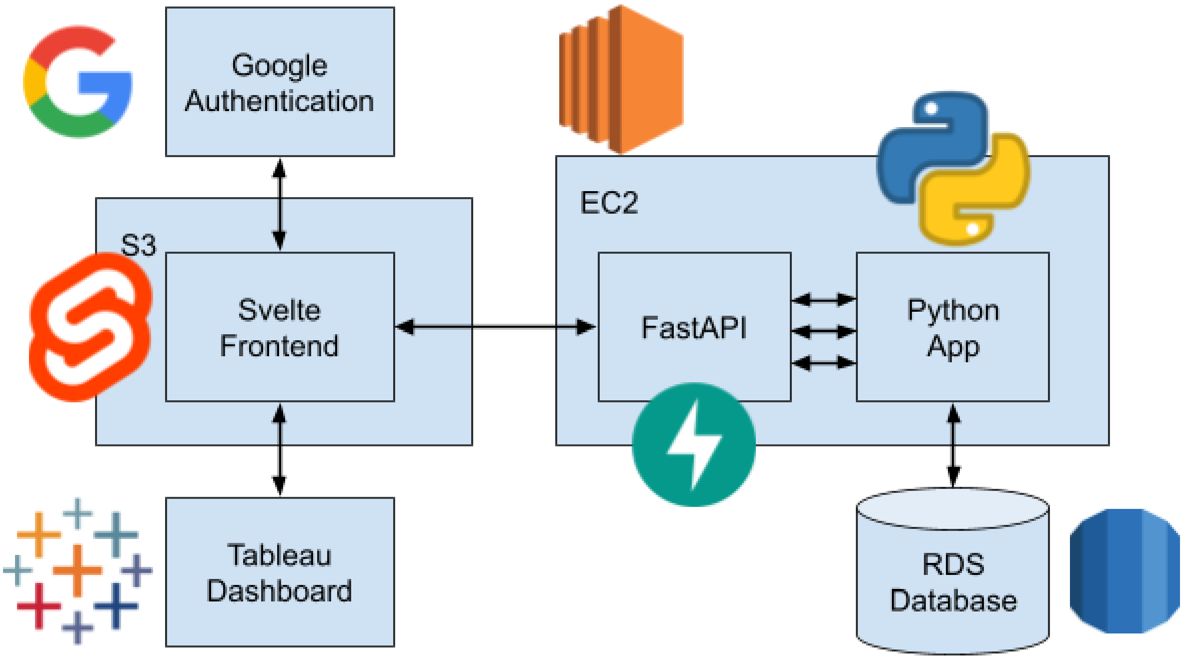

Figure 1: Web Application Architecture

Language

Python	Used for backend system development.
HTML/CSS	For web page structure and styling.
TypeScript	Scripting language for web functionality.
MySQL	Database management language.

Cloud Component

EC2	Virtual server for hosting.
RDS	Cloud-based relational database service for MySQL.

Framework

Svelte	Frontend framework for web page creation.
FastAPI	Backend framework for handling HTTPS requests.

Tool

Git	Version control system for code management.
Bitbucket	Git-based version control platform.
Jira	Task management tool.
Tableau	GUI-based dashboard creation software.
Confluence	Documentation platform

Related Resources

General Information about RDS

Authentication using Google

Svelte