COSC 345 Assignment 1

Study Planner

Team

Luke Hopkirk
Anthony Dong
Aidan Cushman
Sam Merton

For this project, we're developing a study planner application designed to help students manage their academic schedules efficiently. This application will generate a personalised study plan along with their classes, assignments and exams to ensure they're putting enough time into their course.

Feasibility study – Project Plan Evaluation

Technical – Do we have the expertise?

Our group comprises of four third-year Computer Science students, each with 3-4 years of experience in programming. We're all proficient in several programming languages, including Java, Python, R, C, JavaScript, HTML and CSS. While none of us have any prior experience with C++, we are all well-versed in object-orientated programming and C. This knowledge will help facilitate our learning when developing this project. Additionally, we have access to various online resources, as C++ is a very old language with years of documentation. There's also a two-hour lab session every week where we can ask staff for any support.

Operational – Do we have the logistics to make it work?

We'll be storing our repository in GitHub as it provides a robust framework for collaboration within the project. Features like version control, issue tracking and different pipelines will all be useful during development. To track our development progress, we'll use Jira, a powerful suite of tools for agile project management. Jira allows us to implement sprint planning, backlog grooming and burndown charts which are all used in the software development industry. By integrating Jira with GitHub, we can automatically update issues based on commits, pull requests and branch merges. This ensures we're logging everything during the development process. Ultimately, this combination should enhance our workflow, maintaining transparency in task management and issue resolution. Additionally, feature branching in Git allows team members to work on separate features concurrently without affecting the main codebase, promoting parallel development and facilitating smoother merges.

Schedule - Do we have enough time?

When creating a schedule for our project, we used the expect effort estimation in the outlet of planning poker. This works by using a point-based evaluation system for each task, valued by individual group members simultaneously. By scoring each task, we could identify the severity and discuss our reasoning behind it. After this evaluation process, we divided the tasks based on points across different milestones. The early releases have a more fundamental focus with higher points as we must account for feature pushback and more complex user testing in the later stages of development. After our first milestone is complete, we'll come together and conduct a retrospective session to adjust the schedule accordingly.

Resource - Do we have the resources?

Several resources are required to build this project. The first is the IDE, where we'll code; we plan to use Visual Studios Code as everyone in the group has had previous experience with this environment and past projects. Integrated into VS Code, GitHub will be the repository to store our project and track the progress throughout the project. To assist with project management, we're using Jira, which is an online project management software on Atlassian that has all the necessary tools needed to organise tasks, set milestones and monitor the project's overall progress. Regarding our communication as a team, Discord is a perfect fit as it's free, versatile and easy to use.

Requirements

Menu

The study planner's main menu provides a comprehensive overview of the app, displaying essential information such as today's timetable, assignment deadlines, and study reminders. It also allows the student to access various functionalities, including starting a study session, viewing a calendar overview, accessing paper details, and utilising study tools. Students can navigate through these options by selecting the corresponding alphabetical labels on their keyboard, ensuring an intuitive and efficient user experience.

Student profile

A student should be able to create and manage their own profile within the application. This feature would include essential information such as their name, major, field of study, current year, results and the papers they've currently enrolled in. This provides some personalisation for the student. A further extension for this feature would be a sharing functionality where students could export their profiles and share with others.

Adding Papers

A fundamental functionality of a study planner is adding a paper with all relevant details. This feature is crucial as it helps determine the amount of time students should allocate each week to their coursework. By inputting comprehensive information such as paper name, papers points, lectures, labs, tutorials, assignment deadlines and examination dates – along with specific details like day, time and location. The study planner will have all the necessary information to create an academic study schedule tailored for individual students.

Adding assignments

Sometimes, assignment deadlines aren't provided at the beginning of the semester when the paper details are initially set. To address this issue, the app should include a feature to manually input assignment deadlines as they are announced. The assignment due date, alongside dedicated study time, would be dynamically integrated into the student's current study schedule to ensure that enough time and priority are provided to complete the assignment on time.

Adding examinations

Similarly, examination dates are often not available until later in the semester. To accommodate this, the app should offer a feature for adding examination dates once they are released. This functionality would work in much the same way as the assignment feature, allowing students to input exam dates and the location of the exam. As a result of this, their study schedule should automatically adjust to account for these changes.

Displaying paper details

This feature will provide the student with an in-depth summary for the entire paper. It'll include all the essential information, like class times, key dates, upcoming study sessions, and the student's current position in the paper. Having individual paper details in a clear and accessible format can help students quickly review their current position in the specific paper and understand the next steps.

Allocating specific study hours

Academic studies are just one aspect of a student's life. Before creating an effective study schedule, the application must understand how much time can realistically be allocated to studying without impacting other important activities. This feature will allow students to input their available study time per day outside of compulsory classes, and from there, the application can create a study schedule tailored to their schedule.

Generate timetable

This is the core purpose and functionality of the application. After all relevant information has been entered, the app will generate a comprehensive and personalised study timetable. This timetable will integrate the student's classes with dedicated study slots for each paper. Using different paper codes to represent study blocks, deadlines and key dates. This clear and organised layout will help students understand and manage their time, ensuring they stay on track with their academic goals.

Clash detection

Whenever the student's study schedule is updated or refactored, it's crucial that the software can manage scheduling conflicts, as a student cannot be in two places at once or engaging in two different tasks simultaneously. To address this issue, the app will implement a priority scheme based on due dates and the student's current progress for each paper. By assessing these factors, the system can efficiently allocate time blocks, ensuring that tasks are prioritised according to their urgency and importance.

Start study session

This feature allows students to initiate focused study periods with built-in time management tools. When activated, a timer will track the duration of the study block. Additionally, a progress bar will be displayed to help visualise how far through a student is for their current session. To further this feature, students can add study breaks within the session as it's proven to increase productivity compared to studying without stopping.

Study summary

The study summary feature will provide a concise overview of a student's weekly academic progress. This summary will include detailed insights into the time spent studying for each paper, as well as how that time was allocated across the various activities such as class, assignments and examination preparation. To present this information clearly, we plan to implement visual aids such as pie charts, percentages and bar graphs.

Calendar view

Students should be able to see an overview of their schedule in a calendar format. Allowing students to type in a specific date and immediately view their commitments for that day. It will display all scheduled activities, including classes and study sessions while important dates like assignment deadlines and exams should be highly emphasised. The calendar will also incorporate additional features such as reminders and notifications to help ensure that the student is staying on track.

Study tools

Although the main objective of the app revolves around time management, it's equally important to provide students with additional resources to enhance their study efficiency. The app will implement various study tools to assist with learning. These tools include the Pomodoro timer, flashcards, note-taking and quiz generators.

How are we going to build it?

We'll be using the C++ programming language to develop our software, leveraging its robust features and performance capabilities. Our development will take place in Visual Studio Code (VS Code) which is an integrated development environment (IDE) available on macOS, Windows and Linux. Along with this, we will integrate our project into a GitHub repository as it allows all team members to commit their work to different branches of the project. This practice ensures effective version control, collaborating and tracking of changes which are all important aspects in the software development process.

Existing Apps to draw inspiration from

Google Calendar

Google Calendar provides a simple and intuitive UI that displays all necessary features without cluttering the screen. We plan to adopt this approach to create a GUI where users can easily understand and access all functionalities of the app. Like Google Calendar, our app may display the week with a vertical timeline, highlighting different tasks to help keep all information clear and organised for the user.

Notion

Notion is a productivity software that enables efficient and effective time management. We will take inspiration from Notion for several features in our app, such as timetables, assignment scheduling and summary visualisation tools like pie charts. By incorporating these elements, we aim to provide students with a comprehensive and intuitive platform for managing their academic responsibilities.

Anki

Anki is a popular flashcard app that uses spaced repetition to enhance learning and retention. We plan to incorporate a flashcard system like Anki's with spaced repetition algorithms, helping students review and retain information more effectively. This feature will be especially useful for examination and test preparation. Anki also has a very minimal UI approach which can be effective as it simplifies the application.

Evidence that there is customer interest

The growing interest in personal productivity has created a market for tools to assist people in managing their time more efficiently. This trend has become especially important for students, who often must juggle studying, extracurricular activities, part-time work and other personal commitments. In response to this demand, a wide range of apps have emerged to help with time management. Popular apps include Notion, Trello and Google Calendar. This surge in popularity can be displayed through user count as over 500 million people actively use Google Calendar to schedule their day to day. By combining time management features with learning tools, the study planner can target a large audience of students seeking to optimize their academic performance.

How does our project meet the ‘back to the basic’ requirements?

This project embraces the idea of ‘back to the basics’ by choosing C++ as the primary programming language and avoiding third-party libraries. C++ is a well-established language known for its efficiency and simplicity making it perfect to implement our study planner. By focusing on core programming concepts, we’ll have the opportunity to deepen our knowledge on the fundamental principles such as testing, memory management and data structures. Additionally, developing the app in C++ allows us to directly run our program in the terminal which help with both efficiency and resource management.

Risk Management

Technical Challenges

In the event that we are delayed due to unforeseen technical challenges in the C++ implementation, we will revise our timeline and goals to ensure that we still complete the minimum viable product by the deadline. One of the benefits of using the SCRUM ideology is that development will iteratively be completed throughout the sprints. To support this, GitHub’s version control can help us access previous versions of our software if we have technical difficulties closer to the deadline. If our personal devices break for some reason, we also have access to the lab computers, where we can continue developing our software without any major setbacks.

Team Availability

All team members have busy and different schedules meaning that we might not be able to meet all at the same time. To address this, we plan to use the dedicated two-hour lab session to review our weekly sprint while scheduling mini-meet ups throughout the week where we can communicate any issues during the development of our project.

Scope Creep

We are aware that it is easy for a project's deliverables to gradually expand after the beginning of the project which could result in the requirements becoming impossible to complete in the timeframe. To mitigate this, we will describe the project's requirements clearly in our plan and stick to this as much as possible. If we complete all the requirements before the deadline, we will explore more functionalities to add to our software.