

Requirements Document

Uplan

Richard Hartnett, Joe Menzie, Zach Matthews, Steve Lucey

1. Introduction

The software product we are developing is a self-planning planner called UPlan. How it works is first the user will create an account. When creating this account they will fill in certain preferences such as sleep schedule to help the application schedule efficiently based on when the user goes to sleep. The next thing the user will have to do is add events. These events could be classes or other events going on in a specific week. Next the user will enter their assignments as they come in. When entering these assignments they will be asked to enter how long they believe it is going to take, when the due date is, and whether it is a high priority assignment or not. UPlan's algorithm will take these assignments, compare them to the classes or other events the user has for the day, and plan a study schedule based on the free time available. This study schedule will include which assignments to work on that day and for how long. The idea is, the user will use this generated schedule and roughly follow it to keep them on track for their day to day and long term assignments.

The user will be able to edit these events and assignments to alter the time they occur or other settings related to them. One very important aspect the user will need to know for the application to work correctly is at the end of each day or couple days, they will need to go back and update how much time they have left for each assignment. This needs to be done because a lot of time when someone starts an assignment they predict it will take a certain amount of hours but in reality it will take maybe half as long or twice as long. To factor in for this inaccuracy, the user will need to continuously update how far they have come and how much more they have to do.

2. User Requirements

2.1 Software Interfaces

1. Server
 - a. This external system will be used for login of a user. The user will enter their username and password, this will be sent to the server to confirm they have an account, and then the server will either allow or deny the user entrance to the application.
 - b. If the username and password are correct, the server will go into the database and send over the essential user information.
 - c. If the username and password are incorrect, the server will signal for the application to have the user reenter their information or to create an account.
 - d. The server will gather information from the client and relay it to the database to be stored and vice versa.
 - e. The scheduling algorithm will also be on the server. This will minimize the download volume and work handled by the local machine.
2. Java swing and awt library
 - a. These libraries will be used to create the graphical user interface that allows user interaction. Using these libraries, we will design a calendar to display classes, events, and study schedule for that week. This will also include a button to add an assignment, a

- button to add a class/event, a button to edit user preferences, and a button to manually recalibrate the study schedule.
 - b. Java swing and awt libraries will also be used for designing pages to login to an account, to set up an account, to add an assignment/event, to edit preferences, to edit an assignment/event, and to display the a list of all the users assignments/events.
- 3. Java io library
 - a. This library will be used to interact with the user. It will allow the application to store user inputted data. This includes login, adding assignments, creating profile, etc.

2.2 User Interfaces

1. Login/create account
 - a. When the application is opened, the first thing the user will be instructed to do is to enter their username and password. If the username or password is incorrect they will be prompted to enter it again. If it is correct the application will allow entrance.
 - b. If the user does not have an account there will be a create account button that will bring up another page. This page will instruct the user to enter their name, email, password, and special preferences that include sleep schedule.
2. Homepage
 - a. After the user logs in, the first page they will see is the home page. This page includes a visual of a weekly calendar with their classes, events, and suggested study schedule. It also includes several buttons that lead to other pages including add assignment, add class/event, edit account, etc. Of these buttons there will be previous and next week buttons. These buttons will show what the user did the week before and what they have to come in the next week.
3. Add Assignment
 - a. If the user wishes to add and assignment, they will click the add assignment button on the home page and it will bring them to a page where they can enter the information for this assignment. This information includes name of assignment, class assignment is for, due date for assignment, priority for assignment, and estimated completion time for assignment. When this information is entered the user will be instructed to save the assignment.
4. Edit Assignment
 - a. If the user wished to update an assignment, they will double click on the assignment in the calendar and a page similar to the add assignment page will come up. This will be an important part of the application because the user is responsible for updating how much time they have left or is an assignment due date changes. Allowing the user to modify the assignments will allow the application to understand how much time a day the user should be spending on it and will be able to produce the most accurate schedule based on that.
5. Add event
 - a. If the user wished to add a class or event, they will click the add event button on the home page. This will bring up a page to enter information about the event. This information includes day and time, where it is located, if it is a one time event or a weekly event and a priority of the event. When this information is entered the user will be instructed to save the class.
6. Edit event

- a. If the user wished to edit a class or event on the schedule they will double click on it on the calendar. This will bring them to a page similar to the create event page. The user will be able to modify anything they wish and save the changes.
- 7. Edit Profile
 - a. If the user wishes to edit their profile, they will click on the edit profile button on the home page. This will bring them to a page similar to the page used when creating a profile but the information will already be filled in based on what was entered last time it was edited. The user will have the ability to modify their bed and wake time. When the user has finished editing the application will prompt them to save or cancel.
- 8. Assignment List
 - a. If the user wished to view all the assignments they currently have they can click the assignment button. This will bring up a window that has a list of every assignment and all the information that was entered for it.
- 9. Event List
 - a. If the user wished to view all the events they currently have they can click the event button. This will bring up a window that has a list of every event and all the information that was entered for it.
- 10. Manual Recalibration
 - a. If the user has entered a new event or has created a new assignment and wishes to see the their new schedule right then, they will click the manual refresh button on the home page. This button will read adjust the user's schedule and display the new one on the home page.

2.3 User Characteristics

The user base of UPlan will mostly be made up out of college age students. However, it will not be any age restriction in order to use the system. The user will not be required to have any level of technical expertise in order to use the product. The system will require that the user is able to put in the assignments in order for the planner to work optimally. UPlan will be able to be used with minimal experience with the product and technology.

2.4 Assumptions and dependencies

While building UPlan, a few assumptions about the user will be made. This includes the user having a computer with the ability to access and download the program from the internet. The program will require that the user can connect to the internet while being used in order for communication between the user's device and the server. This application also assumes the user has very basic knowledge of how to operate a computer application.

3. System Requirements

These subsections contain all the software requirements at a level of detail sufficient enough to enable designers to design a system to satisfy those requirements, and tests to test that the system satisfies those requirements.

3.1 Functional Requirements

The system must do the following:

- Create a basic calendar
 - Needs to be a weekly calendar with 7 days (broken up by hour of day)
- Allow for user-input
 - Need to be able to enter events/classes
 - Single event from time a to time b
 - Recurring weekly event from time a to time b on given days of the week (ex. classes)
 - Need to be able to enter assignments with due dates
 - Assignments are shown in order of due date in assignments section
 - Allow for breaking up assignment into smaller parts
 - Take in time needed to complete assignment (user predicts this)
 - Represent these events on the calendar
 - Need to be able to enter preferences to allow for better assignment planning
 - Sleep schedule
 - Level of assignment importance
- Create a work plan for assignments and add this plan to the calendar
 - Using schedule, create a plan when to do the assignments in manageable blocks
 - Uses the user's time prediction of each assignment to plan the schedule blocks
 - Display this plan on the calendar with other events
- Update work plan upon user request to accommodate new information assignments and modified assignment time predictions

3.2 Non-Functional Requirements

Performance Requirements:

90% of the responses should be within 10 sec, except for inserting long term blocks into calendar.

Design Constraints:

1. Security: The files in which the information regarding securities and portfolios should be secured against malicious deformations.
2. Fault Tolerance: Data should not become corrupted in case of system crash or power failure.
3. Software Constraints: The system is to run under the UNIX operating system.
4. Hardware Constraints: The system will run on a Ubuntu virtual box with 256 MB RAM, running UNIX. It will be connected to an 8-page-per-minute printer.

External Interface Requirements:

Login Page:

The window will come up in the center of the users screen. The screen will have the Uplan logo at the top. Below it will be two boxes where you will enter your username and password. Below those are two buttons. One will allow you to login and the other to make an account.

Main Page:

The user screen is in the representation of a calendar in the center. The top panel will consist of options such as add, delete or edit assignments. Along with this is the logo in the center of that panel.

4. Future Extensions

UPlan is intended to be a single user software. A possible future extension would be to make it a phone application. Also for now the application requires the user to be connected to the internet when being used so it can connect to the server, but in the future we hope to allow the user to at least access their current calendar without the internet. To recalibrate a schedule, the user would have to reconnect to the internet, but this would make the user's current calendar available anywhere the user has their phone or computer.