Lab 1 Schedule Puzzle Product Outline

Group Gold: Ashley Carter, Jacinda Rosario, Kayla Pineda,
Kofi Amoah, Pablo Castaneda, Xavier Jordan, Ryan Tran, Jake Austin

Old Dominion University

CS 411W, Fall 2023

Professor Thomas Kennedy

August 31, 2023

Version 2

Table of Contents

List of Figures

1. Introduction

- Societal Problem
    - People have difficulty organizing and prioritizing tasks
    - Plans are too rigid and people fail to follow through

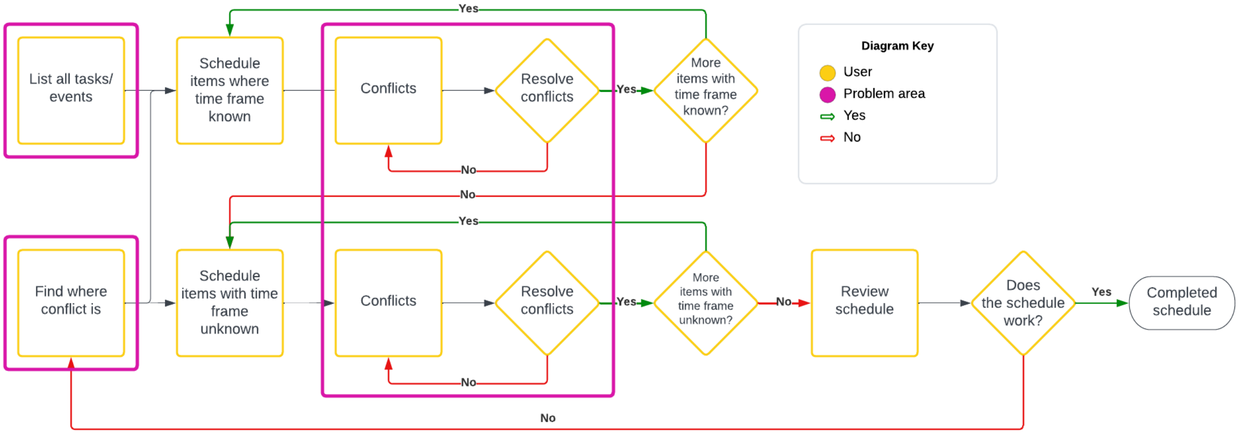


Figure 1: Current Process Flow

- Solution - Schedule Puzzle
    - Web application with automated schedule creation
    - Prioritize selected tasks
    - Semi-automatic conflict resolution

2. Schedule Puzzle Product Description

- Automated schedule creation based on user-supplied tasks
    - User inputs name of task, duration, number of repetitions
    - Schedule Puzzle automatically puts together a schedule based on above
- Basic calendar functionalities
    - Import and export calendars

    - Multiple calendar interfaces
    - Label tasks
    - Notify users of tasks
- Prioritization and customization
    - Prioritization based on categories, deadlines, days, times of days
    - Semi-automatic conflict resolution
    - Custom categories for labeling and prioritization

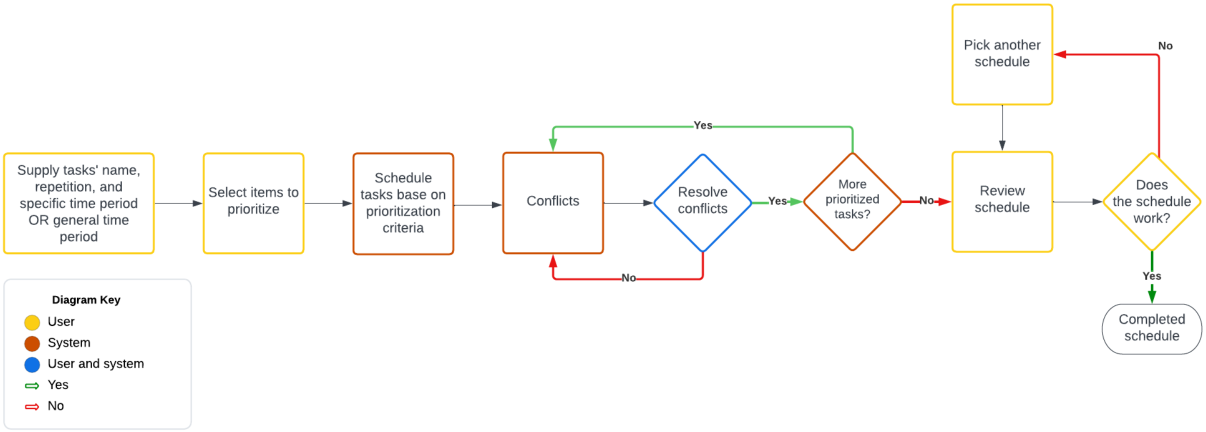


Figure 2: Solution Process Flow

2.1. Key Product Features and Capabilities

- Automate schedule creation based on user-supplied tasks and events
- Unique because it allows for automated schedule creation based on user inputs to all individuals
- This product allows for users to input tasks at the start
- As a result, user is also able to help with conflict resolution
- Alleviate rigid schedule creation by allowing users to work together with the system

2.2. Major Components (Hardware/Software)

- Hardware
    - Personal Computer: Desktop or laptop
    - Cellular Device
    - Tablet
- Software
    - Frontend: HTML, JavaScript, CSS
    - Backend: Python
    - Database: Amazon Web Services, PostgreSQL
    - Framework: Django
    - IDE: VSCode
    - Repository: GitHub

3. Identification of Case Study

- Who is this product for?
    - Individuals who struggle with time management
    - Individuals who already use calendar applications
- Why is this product being developed?
    - Opportunities to be productive are often missed and important tasks are often neglected
    - Alleviate stress that comes with schedule creating
- Who else might use this in the future?
    - Students

- Administration clerks
- Organization leadership
- Starting professionals

4. Schedule Puzzle Product Prototype Description

Feature	Real World Product	Prototype
Basic Calendar Functionalities		
Import existing schedules (.ics, .csv)	Fully functional	Fully functional
Export existing schedules (.ics, .csv)	Fully functional	Fully functional
Has daily/weekly/monthly calendar interface	Fully functional	Fully functional
Modify tasks	Fully functional	Fully functional
Create notes inside of tasks	Fully functional	Fully functional
Send reminders/notifications (push, text, email)	Fully functional	Fully functional
Automation, Customization, and Prioritization		
Automatic schedule creation	Fully functional	Fully functional
Semi-automatic conflict resolution	Fully functional	Fully functional
Custom prioritization	Fully functional	Partially functional
Natural language processing	Fully functional	Partially functional
Behavioral suggestions	Fully functional	Partially/Eliminated

Figure 3: Real World Product vs. Prototype

4.1. Prototype Architecture (Hardware/Software)

- Hardware Utilized
  - Personal Computer
  - Smartphone
- Software Utilized
  - Frontend: HTML, JavaScript, CSS
  - Backend: Python
  - Database: Amazon Web Services, PostgreSQL
  - Framework: Django
  - IDE: VSCode
  - Repository: GitHub
- Prototype MFCD:

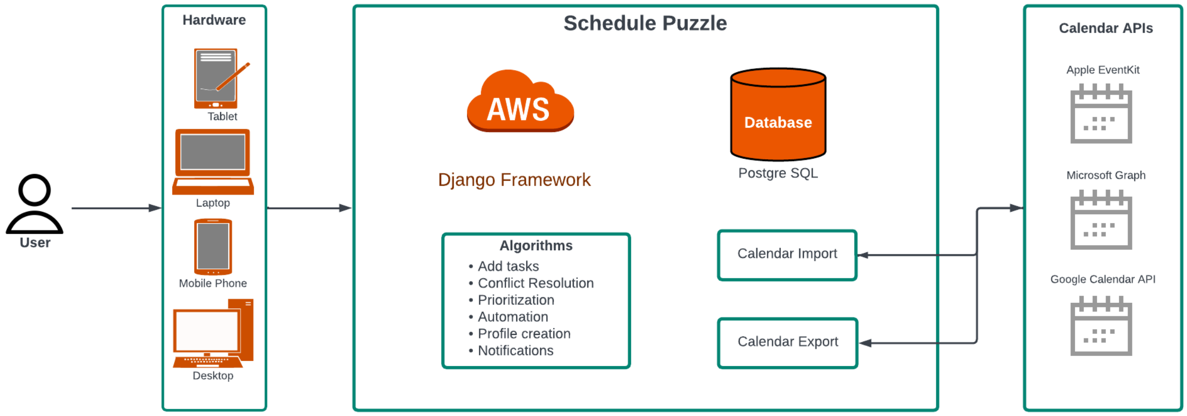


Figure 4: Major Functional Component Diagram

4.2. Prototype Features and Capabilities

- Program demonstrates schedules can be generated quickly and uniquely to meet individual's needs

- Users will be able to create an account to store, edit, and view their various schedules

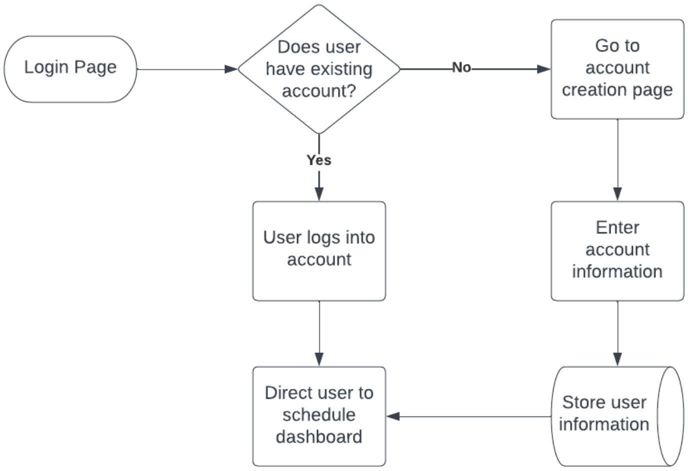

Login Page
Does user have existing account?
No
Yes
Go to account creation page
User logs into account
Enter account information
Direct user to schedule dashboard
Store user information

Figure 5: Profile Creation Algorithm

- Users will have the ability to add tasks by entering in the task fields or by utilizing Natural Language Processing

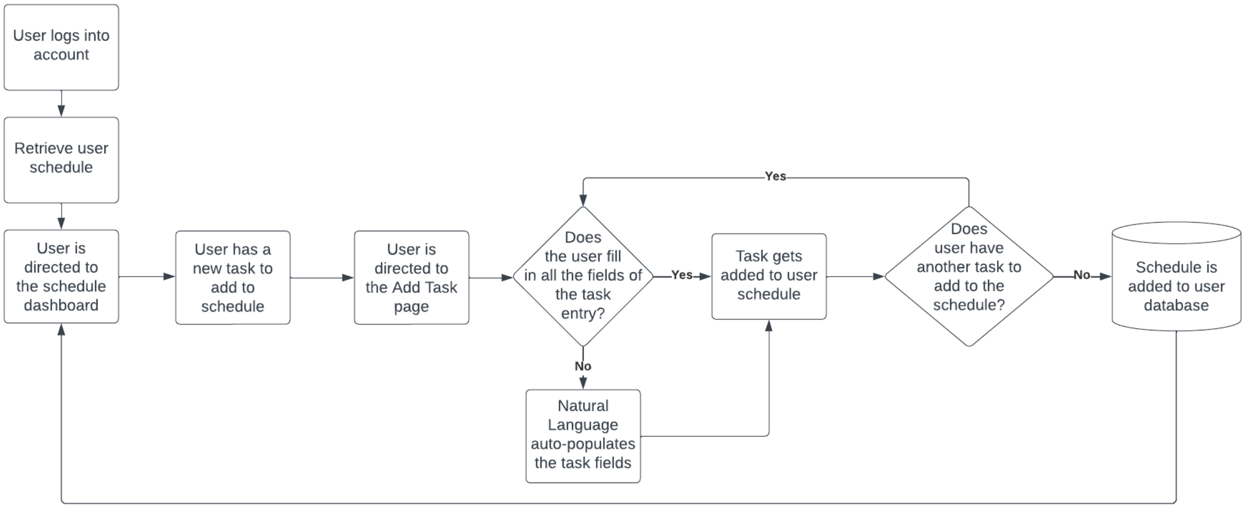

User logs into account
Retrieve user schedule
User is directed to the schedule dashboard
User has a new task to add to schedule
User is directed to the Add Task page
Does the user fill in all the fields of the task entry?
Yes
No
Natural Language auto-populates the task fields
Task gets added to user schedule
Does user have another task to add to the schedule?
Yes
No
Schedule is added to user database

Figure 6: Add Task Algorithm

- Addresses the risks by carefully considering all potential hazards and tackling them with industry best practices

- Helps people create schedules and helps alleviate stress and errors that come with making schedules
- Brings automatic schedule creation to individuals

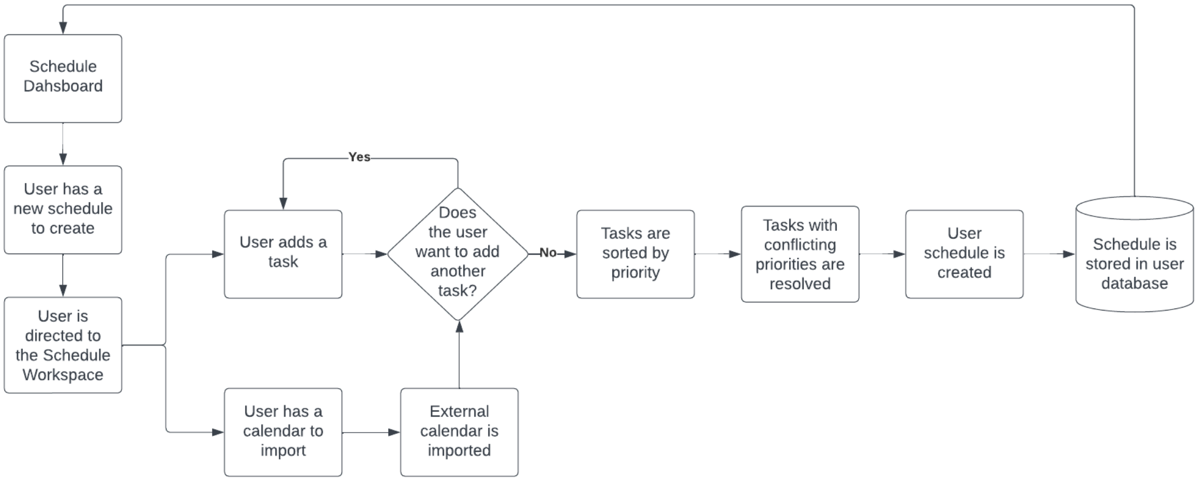

Schedule Dahsboard
User has a new schedule to create
User is directed to the Schedule Workspace
User adds a task
User has a calendar to import
External calendar is imported
Does the user want to add another task?
Yes
No
Tasks are sorted by priority
Tasks with conflicting priorities are resolved
User schedule is created
Schedule is stored in user database

Figure 7: Automated Schedule Creation Algorithm

- Users will have the ability to opt for notifications via Email, text, or push notifications

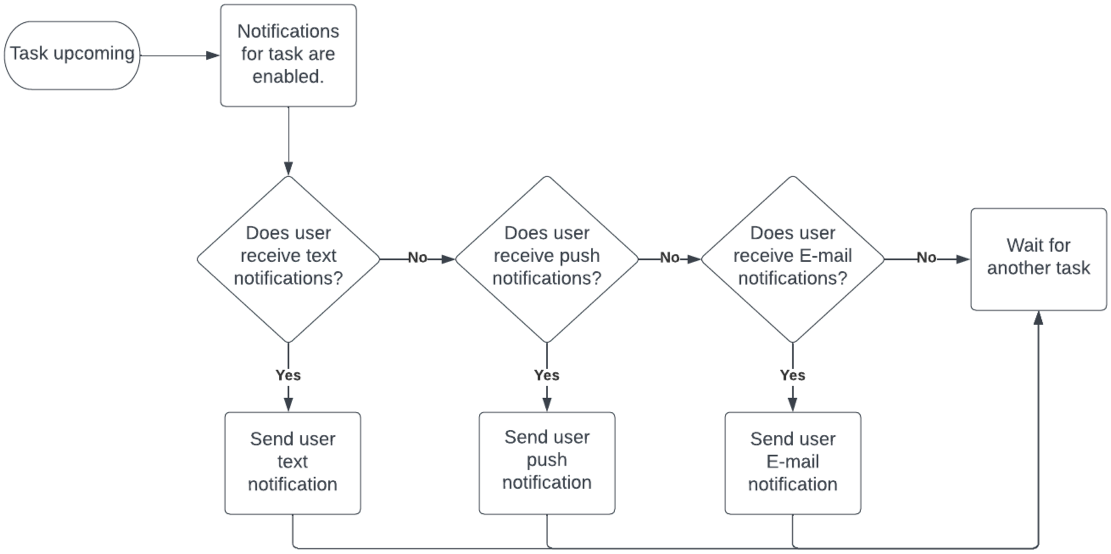

Task upcoming
Notifications for task are enabled.
Does user receive text notifications?
No
Does user receive push notifications?
No
Does user receive E-mail notifications?
No
Wait for another task
Yes
Yes
Yes
Send user text notification
Send user push notification
Send user E-mail notification

Figure 8: Notification Algorithm

- Allows users to import from external calendar applications, such as Apple Calendar, Microsoft Outlook, and Google Calendar

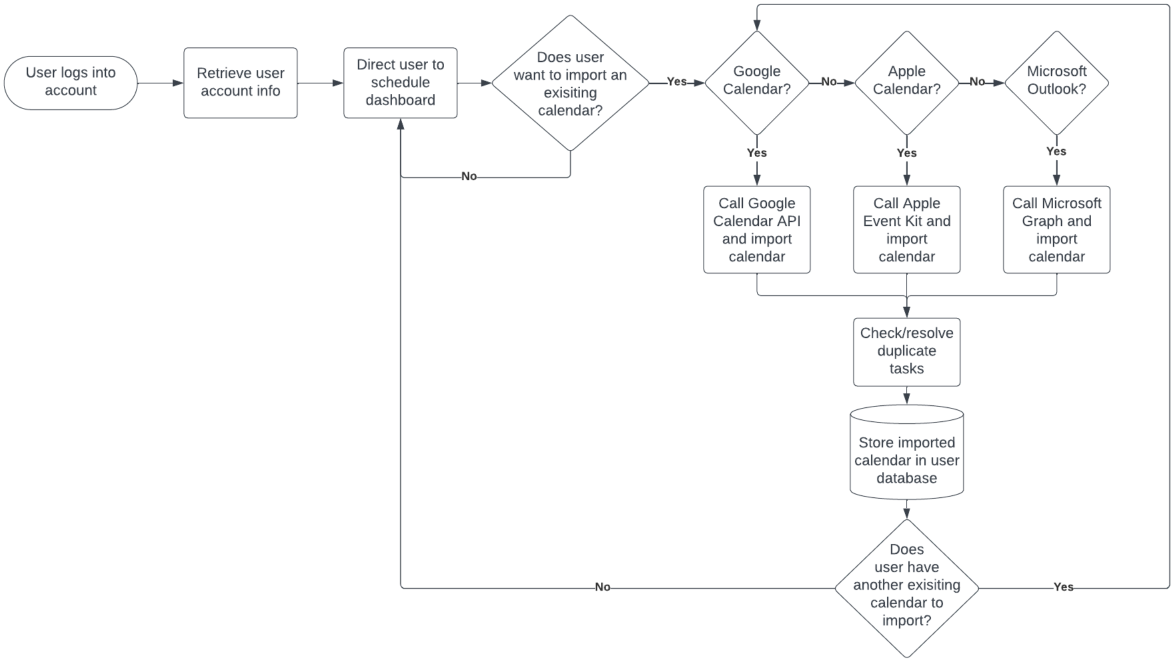

User logs into account
Retrieve user account info
Direct user to schedule dashboard
Does user want to import an exisiting calendar?
Yes
No
Google Calendar?
No
Apple Calendar?
No
Microsoft Outlook?
Yes
Yes
Yes
Call Google Calendar API and import calendar
Call Apple Event Kit and import calendar
Call Microsoft Graph and import calendar
Check/resolve duplicate tasks
Store imported calendar in user database
Does user have another exisiting calendar to import?
No
Yes

Figure 9: Import Calendar Algorithm

- Allows users to export to external calendar applications, such as Apple Calendar, Microsoft Outlook, and Google Calendar

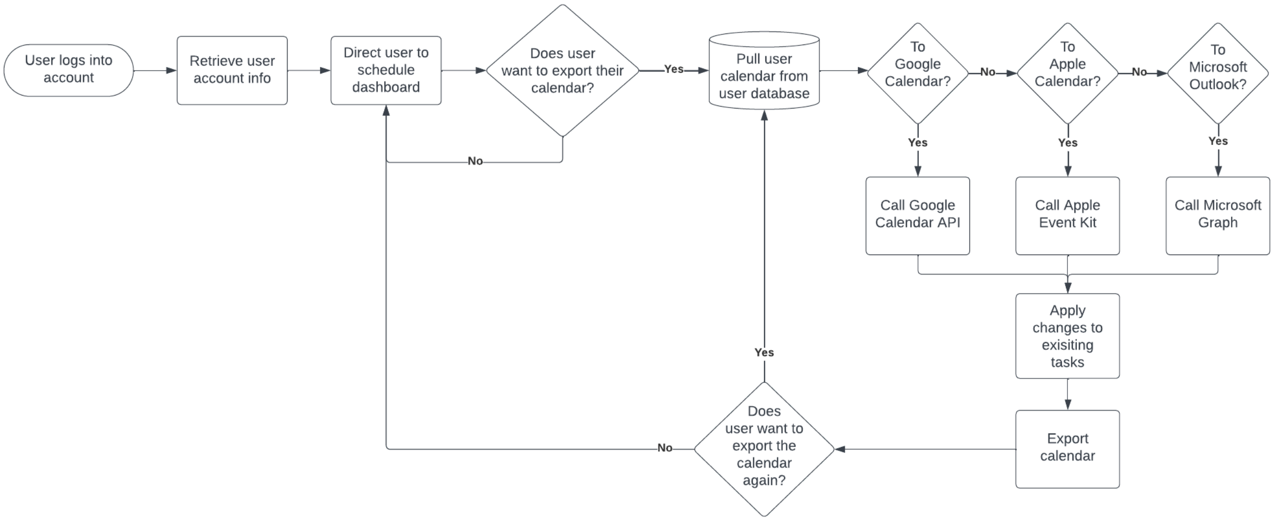

User logs into account
Retrieve user account info
Direct user to schedule dashboard
Does user want to export their calendar?
Yes
No
Pull user calendar from user database
To Google Calendar?
No
To Apple Calendar?
No
To Microsoft Outlook?
Yes
Yes
Yes
Call Google Calendar API
Call Apple Event Kit
Call Microsoft Graph
Apply changes to exisiting tasks
Export calendar
Does user want to export the calendar again?
Yes
No

Figure 10: Export Calendar Algorithm

- Semi-Automatic Conflict Resolution will alleviate any tasks with conflicting priorities

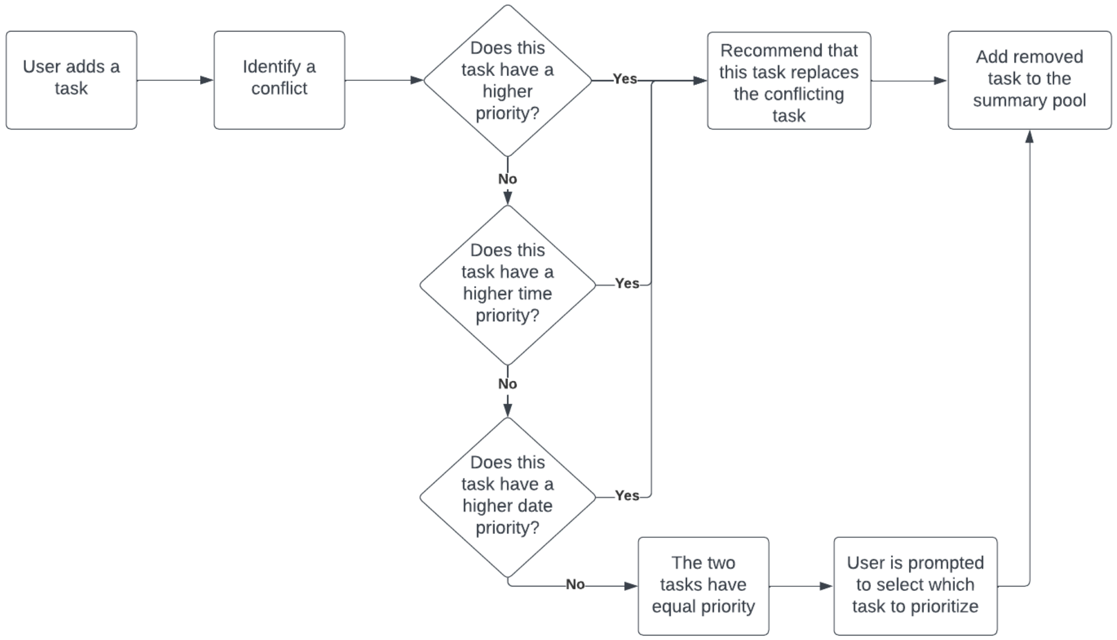
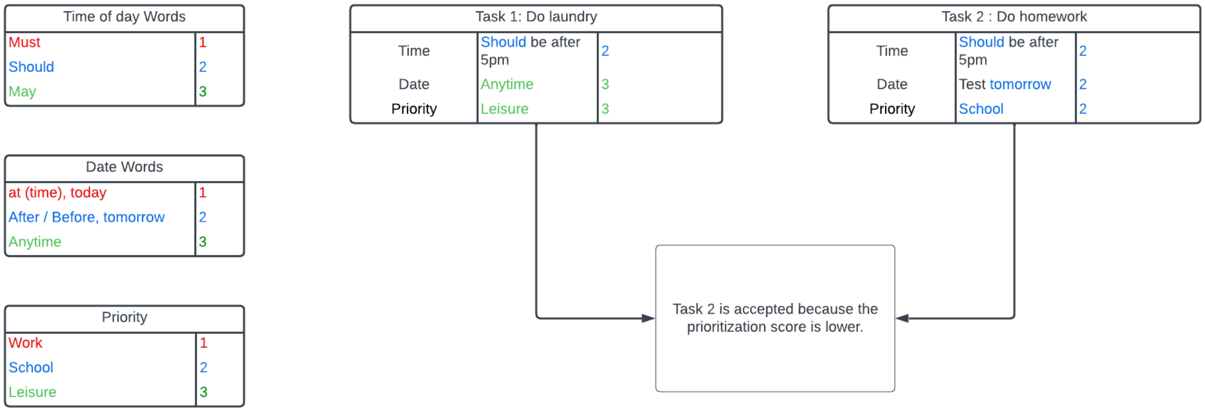

User adds a task
Identify a conflict
Does this task have a higher priority?
Yes
Recommend that this task replaces the conflicting task
Add removed task to the summary pool
No
Does this task have a higher time priority?
Yes
No
Does this task have a higher date priority?
Yes
No
The two tasks have equal priority
User is prompted to select which task to prioritize

Figure 11: Conflict Resolution Algorithm

- Improve the rate at which people accomplish their high priority tasks



Time of day Words	
Must	1
Should	2
May	3

Date Words	
at (time), today	1
After / Before, tomorrow	2
Anytime	3

Priority	
Work	1
School	2
Leisure	3

Task 1: Do laundry		
Time	Should be after 5pm	2
Date	Anytime	3
Priority	Leisure	3

Task 2 : Do homework		
Time	Should be after 5pm	2
Date	Test tomorrow	2
Priority	School	2

Task 2 is accepted because the prioritization score is lower.

Figure 12: Prioritization Example

4.3. Prototype Development Challenges

- Learning the skills to implement the Django framework.
- Learning new languages and API's
- Implementing NLP into our product

Glossary

Task: Used as a catch-all term for things that need to be completed by the user

One time tasks: Events that do not happen regularly as a set time such as appointments or meetings

Recurring tasks: Events that happen regularly at a set time such as school or work.

References

Indeed Editorial Team. (2021, February 22). *12 Time Management Problems (and How To Fix Them)*. Indeed. Retrieved from https://www.indeed.com/career-advice/career-development/time-management-problems

Nemko, M. (2021, December 3). *4 Causes of Poor Time Management | Psychology Today*. Psychology Today. Retrieved from https://www.psychologytoday.com/us/blog/how-to-do-life/202112/4-causes-of-poor-time-management

Prabhu, A. (2022, November 25). *Importance of scheduling tasks and its benefits*. Profit.co. Retrieved from https://www.profit.co/blog/task-management/importance-of-scheduling-tasks-and-its-benefits/

Richardson, B. (2022, October 26). *Time Management Statistics & Facts (New 2022 Research)*. Acuity Training. Retrieved from https://www.acuitytraining.co.uk/news-tips/time-management-statistics-2022-research/