Lab 1 – Schedule Puzzle Product Description

Ryan Tran

Old Dominion University

CS411W

Thomas Kennedy

11 October 2023

Version 1

Table of Contents

List of Figures

1   Introduction

Many people these days struggle with organizing and prioritizing tasks. As time goes on, people tend to have more tasks to deal with while having comparatively less time. A task can be described as anything that needs to be completed, such as one-time tasks or recurring tasks. A one-time task could be an appointment or a meeting; a recurring task could be chores, a job, or school work. Managing tasks is essential to achieving personal and professional life goals; however, many people struggle to even accomplish their daily tasks. According to a 2022 study conducted by Acuity Training, 500 employees were surveyed across several companies. Less than 18% of employees had a proper way to deal with their time management and organization. However, the other 82% mainly used a basic list or their email inbox as an organizational tool. (Richardson, 2022). This means that a majority of working adults across multiple career fields feel that they do not have a sufficient way to organize and manage their tasks.

2   Product Description

Because many people have issues with organizing and managing tasks, people must be able to properly manage their time and organize their tasks in order to complete tasks in a timely manner. If people cannot, then the repercussions could include lost productivity, tardiness, and stress. This is where Schedule Puzzle comes into play. Schedule Puzzle is a web application with automatic schedule creation based on user supplied tasks and events. The application works with the user to create tasks in which then the system prioritizes those tasks and creates a tailored, unique schedule for the user. Schedule Puzzle offers tremendous support to people who want to be more organized through powerful features, such as automated schedule creation, importing and exporting existing schedules, sending reminders and notifications, and prioritizing tasks based on specific variables.

Figure 1 shows the current process flow diagram. The process starts in the top left of our diagram with the user inputting their tasks into the system. At minimum, the user needs to provide the name of the tasks, how often it repeats, and a specific time period or a general time period the tasks occurs. Then, the user would flag the tasks they wish to prioritize. From there, the system takes over. The system would first schedule all the tasks that the user flagged for prioritization. If any conflicts are found, the user is notified of the conflict and decides how to resolve it. Once the conflict is resolved, the next prioritized task is scheduled and the process repeats. The cycle of scheduling and semi automatic conflict resolution applies to each type of task. After the system schedules all the prioritized tasks, the system schedules tasks with a known time frame first, then tasks with a general time frame known, and then tasks that can occur at any time. Once all tasks are scheduled, the system outputs multiple schedules. The user can pick and choose from the output schedules. If the user finds the selected schedule does not work, the user can select another schedule instead of going through the hassle of finding the conflict or completely starting over.

Figure 1

Current Process Flow Diagram
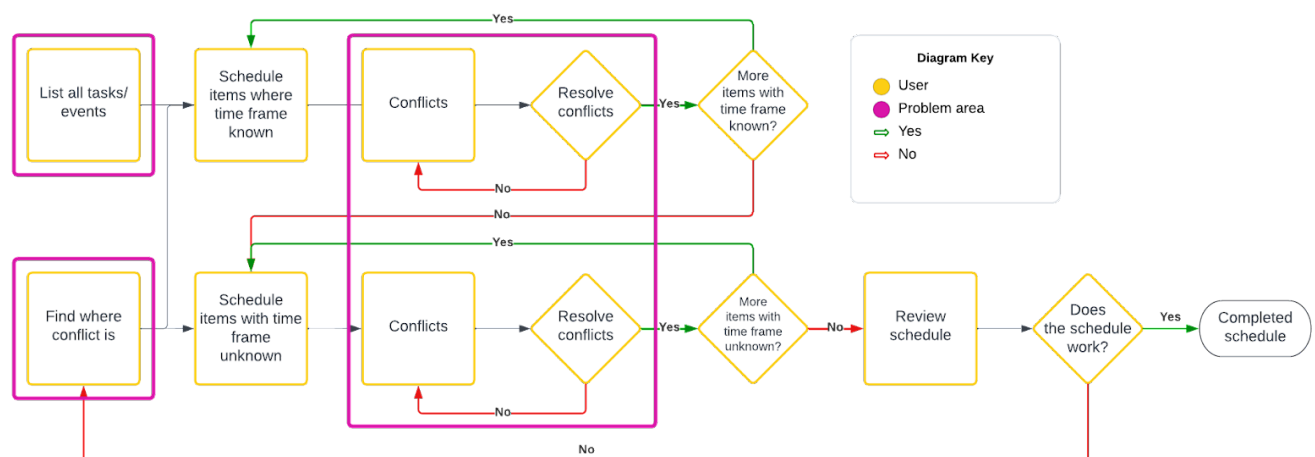
2.1   Key Product Features and Capabilities

The first important feature to introduce is how Schedule Puzzle automates schedule creation based on user supplied tasks. This is split into two parts: the user input and the system automation. The user does the input while the system does the output. The user is able to input the names of the tasks, the duration of each task, and the number of repetitions for each task. Users can create tasks within the Schedule Puzzle system or import calendars from other applications via .ics files or .csv files. Users also have the option to export the schedule into other calendars or spreadsheets. Users can create categories using labels or tags for tasks to further organize the schedule.

The second key feature that Schedule Puzzle introduces is task prioritization. The user is able to flag a task to mark it as a priority. Tasks can also be prioritized based on a deadline; the closer a task is due, the more prioritized it will be on the application. Another prioritization is how long the task would take to complete and the time of day the task needs to be completed. The task can be prioritized too based on the category it is listed under. In the case of two tasks being scheduled together at the same time, the semi automatic conflict resolution algorithm keeps the task with higher priority and pushes back the task with lower priority.

Schedule Puzzle also provides basic features that the typical calendar service provides. The user can import and export calendars via .ics files or .csv files. The user receives a notification through email, text, or push notification when there is an upcoming task. Schedule Puzzle also offers a daily, weekly, and monthly calendar interface for users. Users have the ability to modify tasks and take notes within the tasks.

2.2   Major Component (Hardware/Software)

For the hardware aspect of Schedule Puzzle, the solution consists of a smartphone, desktop, laptop, or tablet. The user uses a tablet, laptop, mobile smartphone, or a desktop computer to add

tasks to their schedule. With those tasks, Schedule Puzzle has algorithms in place to deal with task

addition, conflict resolution, prioritization, automation, profile creation, and notifications. When the

finished schedule is created, the user can import and export the schedule using a Calendar API.

In terms of software, Schedule Puzzle uses Python as the programming language for the main

components and algorithms mentioned above. HTML, JavaScript, and CSS are used for the frontend

web based interface for Schedule Puzzle. For the database, PostgreSQl is used. Django is used for

the libraries. Google Calendar API, Apple EventKit, and Microsoft Graph are used for integrating

the Google, Apple, and Outlook calendars. And externally, Amazon Web Services are used as the

web server host. The IDE used is Visual Code Studio and Github is used for Version Control. As for

group collaboration, the Schedule Puzzle team mainly uses Discord, Google Mail, and Google Drive

to communicate with each other.

3   Identification of Case Study

Schedule Puzzle is targeted at two particular markets. The first customer base consists of

individuals who struggle with time management. Individuals who need help with scheduling and

prioritizing tasks would use Schedule Puzzle to automate a schedule to follow to ensure that the

individual stays productive and manages time efficiently while alleviating stress that usually comes

with schedule creation. The second target market is individuals who already use calendar

applications. If Schedule Puzzle is added as an extension to existing calendar services, this would

allow users to create and save schedules within one system. Individuals that already use calendar

services would use Schedule Puzzle to help refine existing schedules. In the future, consumers of

Schedule Puzzle could be students, administration clerks, organization leaders, or starting

professionals.

4   Schedule Puzzle Product Prototype Description

The Schedule Puzzle prototype is designed to demonstrate the feasibility of importing and exporting existing schedules, having a daily, weekly, and monthly interface, modifying tasks, creating notes within the tasks, and sending reminders and notifications. The Schedule Puzzle prototype also includes automatic schedule creation and semi-automatic conflict resolution. Figure 3 shows the comparisons between Schedule Puzzle's prototype and real world product.

Figure 2

Real World Product vs Prototype Table of Comparison

| 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

4.1   Prototype Architecture (Hardware/Software)

The Schedule Puzzle prototype is mostly consistent with section 2.2 in terms of hardware and software. In figure 3, the major functional diagram shows the hardware used by the user: a tablet, laptop, mobile phone, or desktop. In the prototype, the user adds tasks to the application. The user can also import or export a calendar from an external calendar API. With task creation, the user can input the name and description of the task, the duration of the task, and the repetition of the task. After going through the algorithms, the user is given multiple schedule options to choose from.

Figure 3

Major Functional Components Diagram
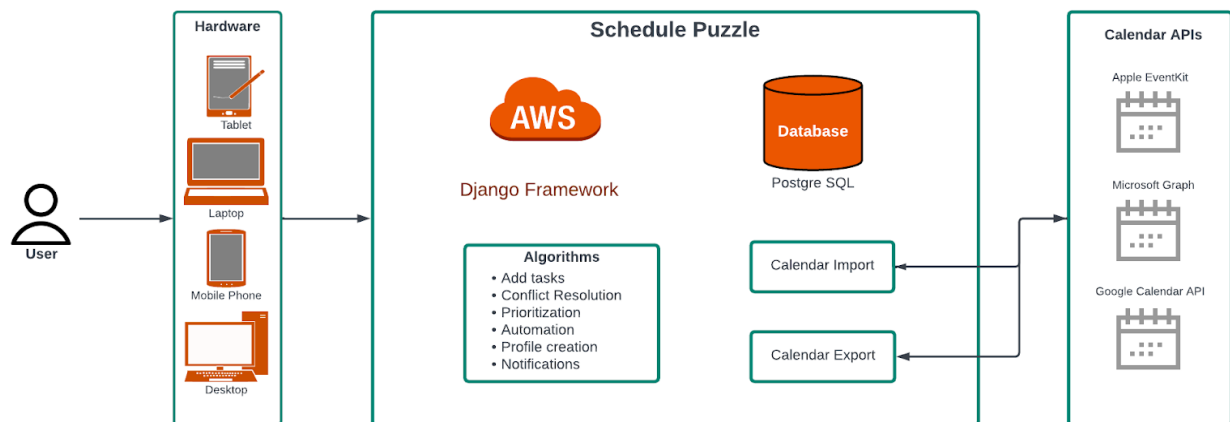
For software, the prototype uses Python as the main programming language for the major components and algorithms. HTML, Javascript, and CSS are used for the front end web based prototype interface. The database used is still Amazon Web Services and PostgreSQL. Framework for the prototype is Django. Google Calendar API, Apple EventKit, and Microsoft

Graph are used for integrating the Google, Apple, and Outlook calendars.The IDE used is

VSCode, and Github is used for version control.

4.2   Prototype Features and Capabilities

As discussed earlier, the major features demonstrated by Schedule Puzzle's prototype are

automatic schedule creation and semi-automatic conflict resolution. The prototype program

demonstrates that schedules are generated quickly and uniquely catered towards an individual's

needs. Both of these major features allow for the program to create schedules and solve conflicts

rather than the user.

The major feature of the Schedule Puzzle prototype is how a program creates an automated

schedule for a user based on the user's inputs. In figure 4, if a user wants to create a new schedule,

the user would start at the Schedule Workspace. The user would need to add tasks with the following

information: the names of the tasks, the duration of each task, and the number of repetitions for each

task. The user also has the option to import a calendar from an external source. Next, tasks are sorted

by prioritization, and any conflicting tasks are resolved. After resolving conflicts, a schedule is

automatically generated for the user. This feature allows for users to relieve stress because the

program automatically creates a schedule for the user. This also makes the user more organized with

daily events, meetings, and other tasks. The system also creates multiple instances of a potential

schedule for the user to choose from.

Figure 4

Algorithm: Automated Schedule Creation
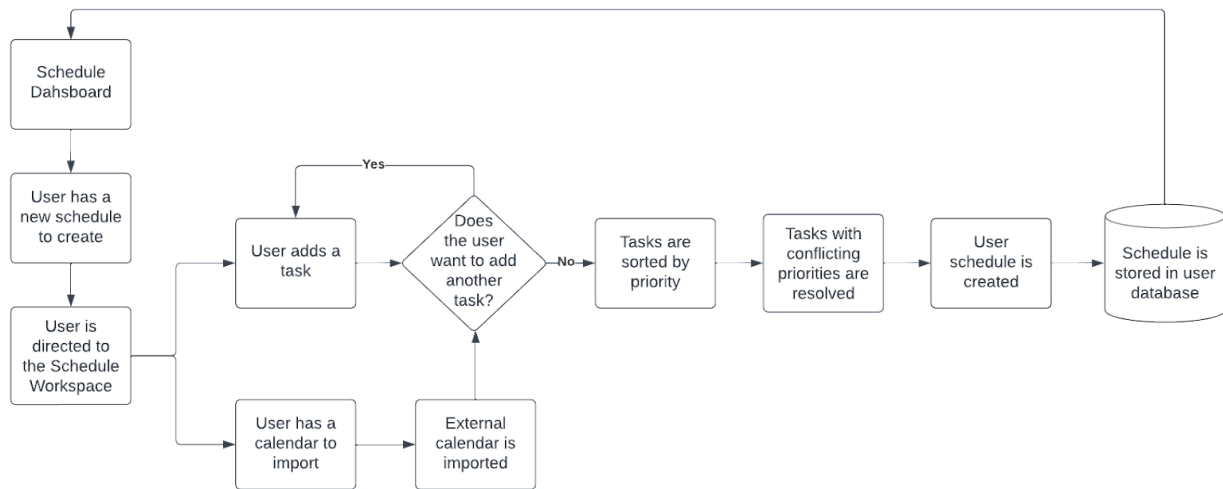
Semi-automatic conflict resolution is how conflicting tasks in the same time frame are resolved. In figure 5, after the user adds a task, there is a conflict check to see if the newly added task has a time conflict with another task already in the schedule. If there is a task conflict, the user is prompted if the current task has a higher priority than the task already in schedule. If the user selects yes, the already scheduled task is put into a summary pool to be looked at later; the new task is added to the schedule. If the user selects no, the user is prompted if the new task has a higher time priority. If the user selects yes, the already scheduled task is put into a summary pool, and the new task is added to the schedule. If the user selects no, the user is prompted if the new task has a higher date priority. If the user selects yes, the already scheduled task is added to a summary pool, and the new task is added to the schedule. If the user selects no as both tasks have equal priority, the user is prompted to choose which task to prioritize and add to the schedule; the unchosen task is added to a summary pool. This feature eliminates the stress for multiple time conflicted tasks as the system continuously prompts the user for which task is more important. After this feature, the program creates an automatic schedule based on the user's input.

Figure 5

Algorithm: Conflict Resolution
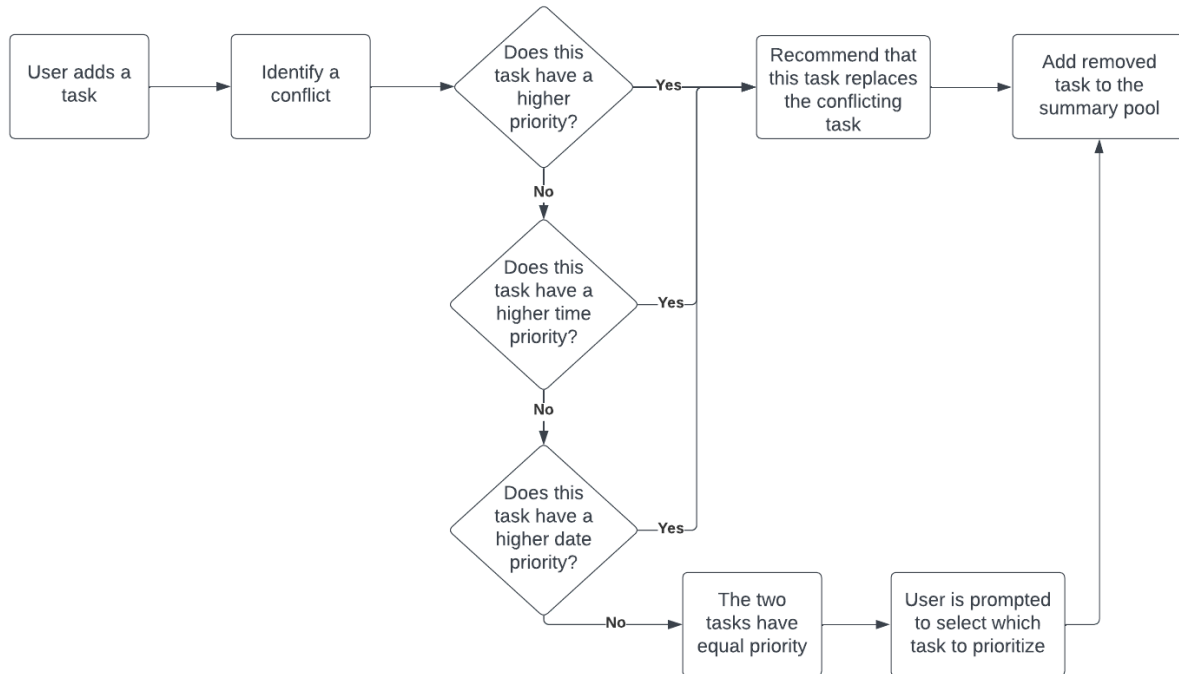
Other features and capabilities of Schedule Puzzle's prototype involve basic features as mentioned in section 2.1. A user is able to create a user profile account in the prototype application. The user can login or sign up for an account if an account does not exist. The user is able to create schedules once logged in with an appropriate account. The prototype demonstrates the importation and exportation of existing schedules via .ics files or .csv files from external calendar applications, such as Apple Calendar, Microsoft Outlook, and Google Calendar. The prototype also includes the option to modify tasks and create notes inside tasks. Users also have the ability to sign up for notifications from email, text, or push notifications for upcoming tasks.

4.3   Prototype Development Challenges

A number of challenges and risks must be overcome during the development of the Schedule Puzzle prototype. The major challenge is learning the skills to create the application for Schedule

Puzzle. The development team must properly learn the skills to implement the Django framework. This also includes learning any new languages and APIs used for the prototype, such as Python, HTML, JavaScript, and CSS. Implementing Natural Language Processing into the prototype is a main challenge as well. Each team mate has to learn any unknown areas of development in order to help create the prototype.

The software developed for the prototype may not meet all the required specifications. This may be due to the inability to complete it within the due time frame. This can be mitigated by dividing the coding portion of the project among the team members. A properly followed schedule is needed to ensure all team members are on task and delegated. The team is using TrelloBoard to keep track of who is doing a specific story. The potential for corrupted or inconsistent data is a risk for the prototype demonstration. Because multiple databases are used, the user information and schedules could be corrupted. Software also has the potential to become corrupt or malfunction. To mitigate this, data backups and testing must be done.

5   Glossary

- **Amazon Aurora:** A part of Amazon Web Services that provides a fully managed relational database engine that is compatible with SQL.

- **Amazon Comprehend:** A part of Amazon Web Services that provides Natural Language Processing to uncover insights and connections in text files.

- **Amazon EC2:** Elastic Compute Cloud is a part of Amazon Web Services that provides servers in Amazon's data centers for building and hosting software systems.

- **API:** Application Programming Interface is a software intermediary that allows two applications to communicate with each other.

- **CSS:** Cascading Style Sheets are a means of describing the presentation of a document written in a markup language. They are typically used to support the display of HTML on web pages.

- **Discord:** A Voice over Internet Protocol (VoIP) and instant messaging social media platform that allows the users to communicate with voice calls, text messages, and sharing files.

- **Django:** A Python-based web application framework that is free and open source.

- **Github:** An online software development platform that is used for storing, tracking, and collaborating on software projects.

- **HTML:** Hypertext Markup Language is a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.

- **IDE:** Integrated Development Environment is a software application that helps programmers develop software code efficiently.

- **Javascript:** A scripting or programming language that allows you to implement complex features on web pages

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

- **PostgreSQL:** An open source, object-relational database built with a focus on extensibility, data integrity, and speed.

- **Python:** A computer programming language often used to build websites and software, automate tasks, and conduct data analysis.

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

- **SQL:** Structured Query Language is a domain-specific language used in programming that is designed for managing data in database systems or stream processing in a stream system.

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

- **Visual Studio Code:** Visual Studio Code is a code editor for building and debugging modern web and cloud-based applications.

6   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-ma

    nagement

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/