Sprint 1 (10/9-11/2)

Space Calendar

Team #22
Members: Sam (Kiara) Grimsley, Audrey Pan, Ella Nguyen, Hart Nurnberg, Yuwen (Reeny) Huang, Lauren D'Souza

#1 Setup Github

Story Point(s): 1 | Name: Kiara

- Set up the team's GitHub repository to centralize collaboration and version control. This included creating the repo, adding all collaborators, and establishing a clear branching strategy.
- The purpose was to give everyone a single source of truth for all project files and ensure that code updates were tracked, reviewed, and merged efficiently.
- While it was quick to complete, this task will be a vital foundation for maintaining organization, preventing version conflicts, and keeping the project workflow consistent across contributors.

#2 Setup Python Environment

Story Point(s): 1 | Name: All

- The entire team collaborated to create a consistent Python environment across all development machines.
- This involved setting up virtual environments, installing shared dependencies, and ensuring that everyone could run the project locally without compatibility issues.
- Establishing this environment early prevented future setup problems, allowing smoother progress and more reliable debugging as the team began implementing core features.

#3 Create UI Draft (Physical)

Story Point(s): 2 | Name: All

- Collaboratively designed a physical sketch of the app's layout, focusing on the user's perspective and flow through different screens.
- The team discussed navigation patterns, key UI elements, and user interactions to align everyone's mental model of how the app should look and feel.

- This rough draft acted as a prototype that guided later digital mockups and influenced the visual direction of the application.

#4 Research: Databases, Hosting, Libraries, etc.

Story Point(s): 5 | Name: All

- Conducted thorough research into suitable technologies for the project, including backend frameworks, databases, and hosting options.
- Compared tools such as Django vs. Flask for backend implementation, and looked into SQLite and PostgreSQL for managing user data.
- The goal was to identify efficient, lightweight, and scalable options that fit the MVP's scope while leaving room for growth in future sprints.
- This research directly informed later architectural decisions, ensuring that every major tool or library used in the project was selected with purpose and understanding.

#5 Create Initial Architecture Document (Except Diagrams & Descriptions)

Story Point(s): 3 | Name: Reeny

- Drafted the initial version of the project's architecture document, outlining the structure, objectives, and key system components.
- The document summarized the app's functionality, technologies, and dependencies, but excluded diagrams and detailed architecture descriptions (added in subsequent tasks).
- This early documentation provided a baseline reference for all team members and made it easier to coordinate roles, ensure clarity, and communicate ideas effectively.

#6 Decide MVP

Story Point(s): 3 | Name: All

- The team collaborated to define the Minimum Viable Product (MVP), focusing on essential functionalities needed to create a usable, demonstrable version of the app.
- Discussions centered on balancing ambition with practicality- deciding which features were critical (like calendar import/export) and which could wait until later sprints.
- Establishing these MVP requirements ensured that development stayed focused and achievable within the project timeline, minimizing feature creep.

#7 Develop Initial Questionnaire

Story Point(s): 2 | Name: Hart & Audrey

- Designed an initial questionnaire aimed at collecting user preferences to customize time slot recommendations.
- Brainstormed questions related to user schedules, productivity patterns, and learning preferences, ensuring the data gathered would later influence personalized scheduling outputs.
- The questionnaire became a key foundation for user input features, bridging the gap between raw calendar data and meaningful schedule suggestions.

#9 Implement Calendar Import/Export

Story Point(s): 5 | Name: Kiara

- Implemented the import/export functionality for calendar files in .ics format, forming one of the core technical components of the app.
- The process involved parsing calendar events, managing data consistency, and testing interoperability with other scheduling tools.
- This feature enables users to seamlessly bring in their existing events and export generated schedules, connecting our system with real-world calendar applications.

#16 Create Sample Data

Story Point(s): 2 | Name: Reeny

- Created several sample .ics calendar files containing mock events to serve as test data for development and debugging.
- The files represented different user scenarios and workloads, helping the team validate import/export performance and simulate realistic app interactions.
- This sample data was essential for early-stage testing before real user inputs were available, ensuring that new features could be verified immediately.

#18 Create Reference Stories

Story Point(s): 2 | Name: All

- Developed a set of reference user stories to serve as baselines for estimating story points in future tasks.
- Each reference story captured a clearly defined task with known difficulty, making it easier to assign consistent point values later.

- This exercise helped improve estimation accuracy, promoted fairness in workload distribution, and reduced uncertainty during sprint planning meetings.

#22 Testing Features

Story Point(s): 5 | Name: All

- ln this initialization stage, the team performed light testing focusing on verifying that basic components, such as import/export functionality and early UI features, worked correctly.
- Most testing was manual, aimed at catching early bugs and confirming system stability during setup.
- As development progresses, testing will become more involved, expanding to include structured unit, integration, and end-to-end tests to maintain quality as new features are added.

#25 Setup Website

Story Point(s): 5 | Name: Kiara

- Developed the initial skeleton of the website using Django, establishing the project's foundational structure.
- This setup included creating necessary folders, initializing routing, and linking the front-end and backend for future integration.
- The website skeleton served as a launch point for subsequent sprint work, allowing the team to immediately begin building and testing real features in a cohesive environment.

#43 Create UI UML Diagram

Story Point(s): 5 | Name: Ella

- Designed a UML diagram mapping the overall user flow of the app, showing how users navigate between screens and interact with features.
- The diagram provided a visual understanding of the system's behavior and helped ensure design consistency between the UI and underlying logic.
- This artifact was valuable for both documentation and team communication, aligning developers and designers on how the application should function from a user perspective.

#44 Create Import/Export UML Diagram

Story Point(s): 5 | Name: Audrey

- Created a UML diagram specifically for the import/export process, visualizing how the system reads and writes .ics calendar files.
- The diagram mapped out the data flow and key interactions between components, clarifying how data moves from file to program and back.
- This visualization helped the team identify dependencies early and served as an implementation guide for backend development.

#45 Create UML State Diagram

Story Point(s): 5 | Name: Lauren

- Created a detailed UML state diagram illustrating how the website transitions between states based on user actions and inputs.
- The diagram depicted possible user interactions and system responses, showing how the site maintains consistent behavior under different scenarios.
- This helped developers understand how dynamic elements of the app should react, reducing ambiguity in implementation and debugging later on.

#46 Architecture Summary

Story Point(s): 2 | Name: Hart

- Wrote a comprehensive summary of the app's overall architecture, tying together the front-end, back-end, and data flow components.
- The summary included a discussion of technologies used and how data travels through the system.
- This write-up provided a big-picture overview that will help the team quickly understand how all parts of the project fit together.

Sprint 2 (11/2-11/16)

Space Calendar

Team #22
Members: Sam (Kiara) Grimsley, Audrey Pan, Ella Nguyen, Hart Nurnberg, Yuwen (Reeny) Huang, Lauren D'Souza

#25 Set Up Website

Story Point(s): 5 | Name: Kiara

- Expand the existing Django website skeleton with new functionality, including dynamic views, and improved navigation
- Create a smoother and more interactive user experience, integrating pages for preferences, task management, and calendar
- Establishing these foundational pages will make it easier to connect the front-end UI with the back-end logic and prepare the platform for user testing later on

#37 Add "Add Task" Button + Form

Story Point(s): 2 | Name: Ella

- Introduce an interactive "Add Task" button and accompanying form that allows users to input new tasks into the system
- Users will be able to enter various data fields such as title, duration, priority, which the program will later use to populate their calendar
- One of the main user interaction points, bridging user input with automated scheduling functionality

#11 Initial Calendar Space Fill

Story Point(s): 3 | Name: Hart

- Implement the logic that fills available calendar time slots with user tasks
- Take existing events into account and intelligently assign new tasks to open periods in the user's schedule
- Establish core scheduling engine that powers the app's main purpose (automatically organizing a user's study or work time)

#10 Create Preferences (Toggle)

Story Point(s): 3 | Name: Audrey

- Define and implement an initial list of user preferences, such as study length, time-of-day preferences, and productivity windows
- Preferences may be toggled or selected from menus, giving users greater control over how the scheduling algorithm prioritizes their time
- Make the app feel more personalized and adaptable to different working styles

#8 Set Up Data Storage

Story Point(s): 8 | Name: Reeny

- Implement reliable data storage, using either a SQL database or browser-based storage solutions such as cookies or local storage
- Store key information like user preferences, tasks, and calendar data between sessions
- Set up persistent data storage to enable more complex features in later sprints

#15 Set Up Table of Event Categories

Story Point(s): 2 | Name: Reeny

- Create a structured table or database schema for event categories, such as “Study,” “Break,” or “Meeting”
- Categorization will make it easier to filter, color-code, and prioritize events in the UI and scheduling logic
- Will help support future preference matching, and helps make the calendar more visually organized and user-friendly

#17 Existing Event Parsing

Story Point(s): 5 | Name: Lauren

- Develop logic that scans imported calendar files for pre-existing events
- Identify busy and free periods, ensuring new tasks aren’t scheduled during conflicts
- Will help generate realistic and non-overlapping schedules

#28 Error Logging & Crash Reporting

Story Point(s): 2 | Name: Audrey

- Introduce basic error handling and crash-reporting mechanisms to capture unexpected issues during runtime
- Create readable logs that help developers trace and fix problems efficiently without interrupting user experience

- Streamline debugging and improve application stability as more complex features are added

#30 Timezone & Time-of-Day Handling

Story Point(s): 2 | Name: Kiara

- Manage time zones and restrict scheduling to reasonable or user-preferred hours of the day
- Ensure event times remain accurate across different regions and daylight-saving changes
- Improve scheduling precision so tasks align naturally with each user's working or study hours

#31 Recurring Events

Story Point(s): 3 | Name: Hart

- Allow users to create recurring events that automatically repeat on multiple days at consistent times
- Include flexible options for daily, weekly, or custom recurrence intervals
- Strengthen the app's long-term scheduling capabilities and reduce repetitive manual input for users

#39 Display Event List Under Calendar

Story Point(s): 2 | Name: [TBD]

- Display a list of upcoming events directly beneath the calendar view
- The list will include event titles, categories, and times to give users a quick overview of their schedule
- Make it easier to track and adjust scheduled tasks without leaving the main calendar screen

#19 Create Native Calendar UI

Story Point(s): 8 | Name: [TBD]

- Introduce an in-app native calendar interface that lets users visualize their schedules directly within the application
- Display both imported and newly generated events, updating dynamically as users add, remove, or modify tasks
- Provide an intuitive and interactive layout that improves usability and reduces reliance on external calendar tools

#22 Testing Features

Story Point(s): 8 | Name: All

- Expand testing coverage as new sprint features are introduced, including data storage, task creation, and scheduling logic
- Begin incorporating unit and integration tests to verify system reliability before deployment
- Emphasize early bug detection and quality assurance as the project transitions from setup into active feature development

#47 Artifacts Document

Story Point(s): 5 | Name: Ella

- Compile all sprint stories, descriptions, and planning details into a formal artifact document
- Maintain consistency with past sprint documentation while focusing on upcoming goals rather than completed work
- Ensure clear communication of sprint objectives, priorities, and rationale for each story

Sprint 3