CS3354 Software Engineering
Final Project Deliverable 2

FocusQuest

Group 1
4-29-2025

1. **Project Goal/Motivation/ Task Delegation for 1st and 2nd deliverables**

    1. Group Members:
        ○ Sanskriti Aripineni
        ○ Javier Benzo San Martin
        ○ Ndubueze Echefu
        ○ Rapheal Evangeslista
        ○ Leslie Flores
        ○ Steven Martinez Sanchez
        ○ Disha Shetty
        ○ Hunter Smith
        ○ Juan Pablo Laseter

    2. Proposed Implementation
        ● FocusQuest is a mobile/web application designed to gamify productivity, track focus sessions, and provide AI-driven insights to enhance users' work and study habits. The app helps users understand their most productive times by analyzing focus patterns and offering personalized feedback.

           Users earn points, streaks, and levels based on task difficulty, encouraging habit-building and motivation. Additionally, an AI-powered chatbot is implemented to provide encouragement, focus tips, and personalized suggestions to ease the burden of hectic tasks and to break them down into subtasks. FocusQuest operates beyond work and study, it can be used to track productivity across various activities such as working out, attending classes, and cooking, since the main idea of this application is to reinforce positive habits in all aspects of life to prevent doom scrolling and excessive screen time.

    3. Motivation
        ● We are in a world where everything goes very fast. Staying focused is now harder than ever. Traditional productivity methods are hard to follow, Is very difficult to build lasting habits. FocusQuest tries to tackle this by combining gamification, AI-driven insights, and multi-activity tracking to make productivity attractive and personalized to everyone's needs.

           This project is an exciting challenge in terms of software engineering, requiring mobile/web development and the use of AI. FocusQuest helps users stay motivated and focused by taking advantage of technology.

4. Audience
   ○ FocusQuest is designed for a diverse audience with a goal of enhancing their productivity and developing lasting habits in an engaging way. Students can use this app to stay on track with their academic work, while professionals can use FocusQuest to structure their tasks effectively around project management and team morale. FocusQuest reaches out to students and professionals who share an enthusiasm for gamification, self-improvement, and developing good habits. It offers personalized optimization of multi-task management through the lens of a video game. In essence, whether you're a student or a professional, or just someone who is willing to take productivity to the next level, FocusQuest is here to help.

2. Link to our repository: [3589hunter/3354-Group-1](#)

3. Delegation of tasks:
   Deliverable 1:

Name	Delegated Task
Disha Shetty	Creating Requirement specification document (including functional and non-functional requirements), completing step 1.1-1.2, task delegation and management, document submission
Leslie Flores	Creating Use case diagrams
Sanskriti Aripineni	Creating Use case diagrams, Traceability matrix
Rapheal Evangeslista:	Creating Sequence Diagrams for Use Cases
Javier Benzo San Martin	Creating the class diagram
Steven Martinez Sanchez:	Creating the architectural design and describing the reason for the architectural pattern design.
Hunter Smith	Creating git Repository, creating class diagram, outlining the classes that will be used for the project
Ndubueze Echefu	Creating the architectural design and describing the reason for the architectural pattern design
Juan Pablo Laseter	Creating Sequence Diagrams for Use Cases

Deliverable 2:

Name	Delegated Task
Disha Shetty	Deliverable 1 Requirement update: API Figma prototype, Task management, implementation using replit, demo slides. conclusion, FocusQuest implementation document, FP calculation
Leslie Flores	Create software test plan
Sanskriti Aripineni	Deliverable 1 update Scalability Slides part 2
Raphael Evangelista:	Similar design concept and comparison
Javier Benzo San Martin	Create software test plan
Steven Martinez Sanchez:	Cost effort and estimation, Similar design concept and comparison, FP calculation
Hunter Smith	Deliverable 1 update Git repo, Slides part 1
Ndubueze Echefu	Project scheduling
Juan Pablo Laseter	Project duration and staffing

4.   Software Process Model

Waterfall model: We have opted to implement the Waterfall model for this project due to the well-defined requirements and the strict timeline associated with it. Given that this is a class project with limited time, we believe the Waterfall approach will provide the necessary structure and discipline, ensuring that each phase is completed before moving on to the next. Additionally, as our objective is to deliver a final presentation at the conclusion of the project, the sequential nature of the Waterfall model aligns well with our goals, minimizing flexibility towards changes and providing a clear, organized framework for successful project completion.

5.   Requirement Specification

1. Introduction to Document
   a. Purpose of Product
      i.   FocusQuest is a mobile/web application designed to gamify productivity, track focus sessions, and provide AI-driven insights to enhance users' work and study habits.
      ii.   The app shall help users understand their most productive times by analyzing focus patterns and offering personalized feedback.
   b. Scope of Product
      i.   Users shall earn points, streaks, and levels based on task difficulty, encouraging habit-building and motivation.
      ii.   Additionally, an AI-powered chatbot shall be implemented to provide encouragement, focus tips, and personalized suggestions
      iii. To ease the burden of hectic tasks it shall break them down into subtasks.
   c. Acronyms, Abbreviations, Definitions
      i.   Acronyms
            1. AI: Artificial Intelligence
   d. Definitions
      i.   Streaks: Running count of consecutive high productivity days

2. General Description of Product
   a. Context of Product
      i.   To help stay focused in this fast paced world.
      ii.   Traditional productivity methods are hard to follow, and it is very difficult to build lasting habits.FocusQuest shall tackle this by combining gamification, AI-driven insights, and multi-activity tracking to make productivity attractive and personalized to everyone's needs.
   b. Product Function

        i.   Gamify productivity through streaks and points.

        ii.   Assign levels based on task difficulty level.

        iii.   Use of AI chatbot to streamline tasks and provide personalized tips and suggestions based on user data and history.

c. User Characteristics

        i.   Students

               1. Possible use: To stay on track with their academic work

        ii.   Professionals

               1. Possible use: To structure their tasks effectively around project management and team morale building.

d. Constraints

        i.   For AI personalized feedback to be useful, users would have to use the app for 3-4 months.

e. Assumptions

        i.   It is assumed that the user has basic technological experience in using mobile/web application

f. External Dependencies

        i.   The app will use Google Gemini API to provide personalized feedback

g. Scalability

        i.   We can have the application backend supported on a cloud platform. We can use AWS, Azure, or Google Cloud. This will ensure auto-scaling based on real-time traffic and help the app handle a growing number of users.

3. Specific Requirements

    a. Functional Requirements

        i.   Login

1. Shall allow customers to login to an account to ensure their data is safe, and previous tasks and streaks are continued.
2. Inputs: Users shall login with username and password
3. Processing: Shall validate based on existing database if existing user or shall add user information to database.

 ii. Manage Focus Sessions
  1. Shall allow users to track sessions
  2. Shall measure and analyze focus duration

 iii. Create tasks
  1. Shall allow users to add tasks they want to complete.
  2. Should allow users to add date/time by when they want to complete it by
  3. Shall allow users to enter expected time for task

 iv. Earn Points
  1. Points shall be assigned on task completion
  2. 1 Point/minute according to the expected time on task will be assigned, to measure difficulty.

 v. Manage Tasks
  1. Shall allow users to modify information on added tasks, such as expected time, due date/time
  2. Shall allow user to remove/replace task

 vi. AI feedback
  1. Shall provide encouragement to complete tasks.
  2. Should provide personalized tips and strategies based on user data
  3. Should provide personalized peak productivity time frames for different types of tasks according to past user data

b. **Non Functional Requirements**
 i. Availability

1. Occasional maintenance breaks expected but shall be no longer than 1 hr.
2. Shall not be available offline

ii. Security
1. User data is secure and databases shall not be accessible unless legally required.

iii. Usability
1. The interface shall be easy to use and anyone with basic technology experience should be able to use it with ease.

iv. Performance
1. Shall not take more than 10 seconds to login.
2. Task tracking and AI feedback should not take more than 90 seconds.

5. Diagrams (Use-Case/Sequence/Class/Activity

5.1.   Use case diagram
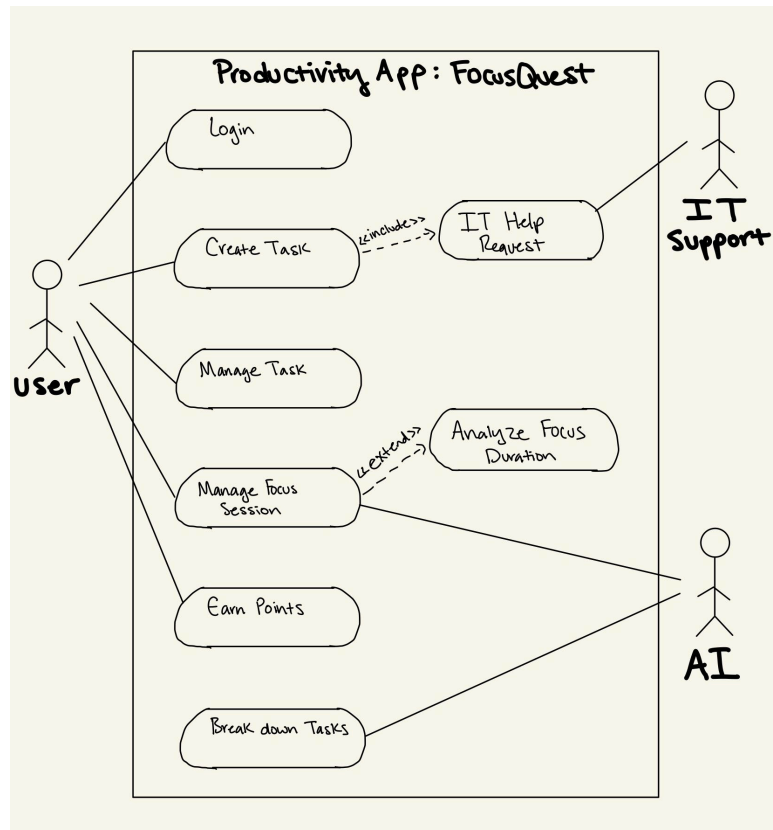
5.2.   Traceability Matrix

Use Cases
- Login
  1. Login
  2. Logout
  3. Create account
  4. Reset password
- Create task
  5. Add task
  6. Delete task
- Manage task
  7. View report
  8. Edit task
- Earn points
  9. Streak

10. Points
- Break down
11. AI feedback
12. importance
13. deadline

Functional Requirements and Use Case

Req	Priority	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13
R1	1	x												
R1.1	1	x												
R1.2	1	x		x										
R1.3	1			x										
R1.4	1	x	x		x									
R2	2					x	x							
R2.1	3							x						
R2.2	1							x	x					
R3	1					x								
R3.1	1					x								
R3.2	1					x	x							
R3.3	2						x							
R4	2									x				
R4.1	2									x	x			
R4.2	1										x			
R5	3			x		x			x					
R5.1	2					x		x						
R5.2	1						x		x					
R6	1											x		
R6.1	1											x		
R6.2	1											x	x	x
R6.3												x	x	
UC priority		1	1	2	3	1	1	2	2	3	4	1	1	1

Non-Functional Requirements and Use Case

NFR	Priority	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13
NFR1	1	x												
NFR1.1	1	x	x	x	x									
NFR1.2	2	x												
NFR2	1	x												
NFR2.1	1					x								
NFR3	3					x	x	x	x					
NFR3.1	3					x	x	x	x					
NFR4	4	x												
NFR4.1	4	x												
NFR4.2	3											x	x	x
UC priority		1	1	2	3	1	1	2	2	3	4	1	1	1

Non-Functional Requirements and Functional Requirements

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	Req	Priority	FR1	FR1.1	FR1.2	FR1.3	FR1.4	FR2	FR2.1	FR2.2	FR3	FR3.1	FR3.2	FR3.3	FR4	FR4.1	FR4.2	FR5	FR5.1	FR5.2	FR6	FR6.1	FR6.2	FR6.3
	NFR1	3																						
	NFR1.1	3																						
	NFR1.2	3																						
	NFR2	1			x																			
	NFR2.1	1	x	x		x	x																	
	NFR3	2									x									x					
	NFR3.1	2				x	x	x	x			x	x	x	x	x	x	x	x	x			x	x	x
	NFR4	1																							
	NFR4.1	2	x	x	x	x	x																		
	NFR4.2	1									x							x				x	x	x	x
	FR Priority		1	1	1	1	1	3	3	2	1	1	1	1	2	2	1	2	1	1	1	1	2	2

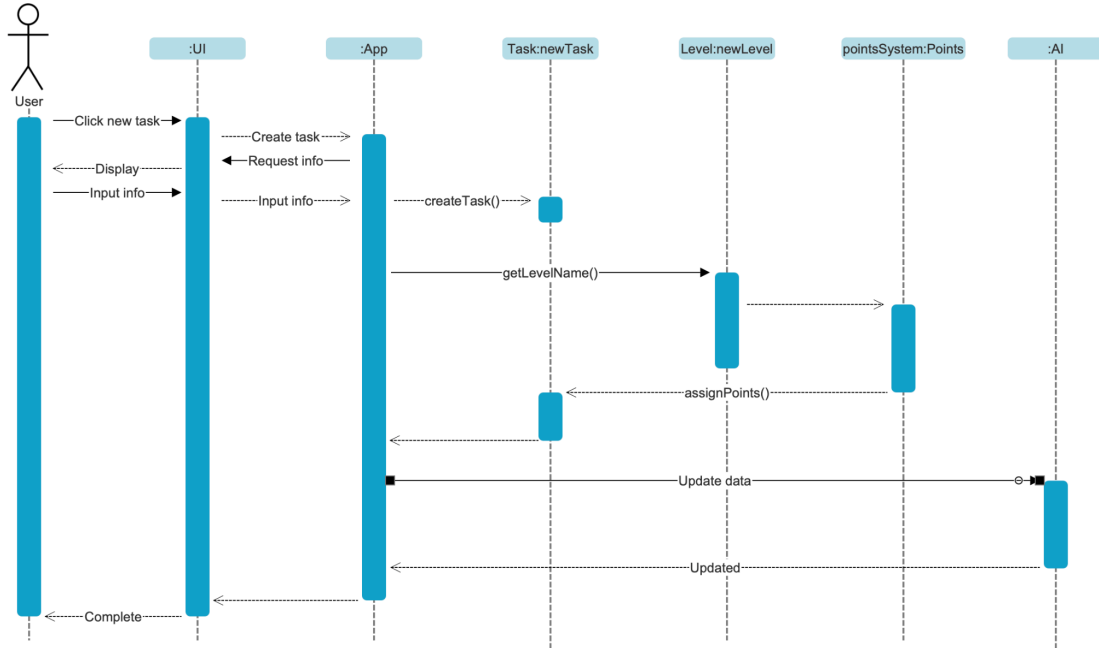## 5.3. Sequence diagram
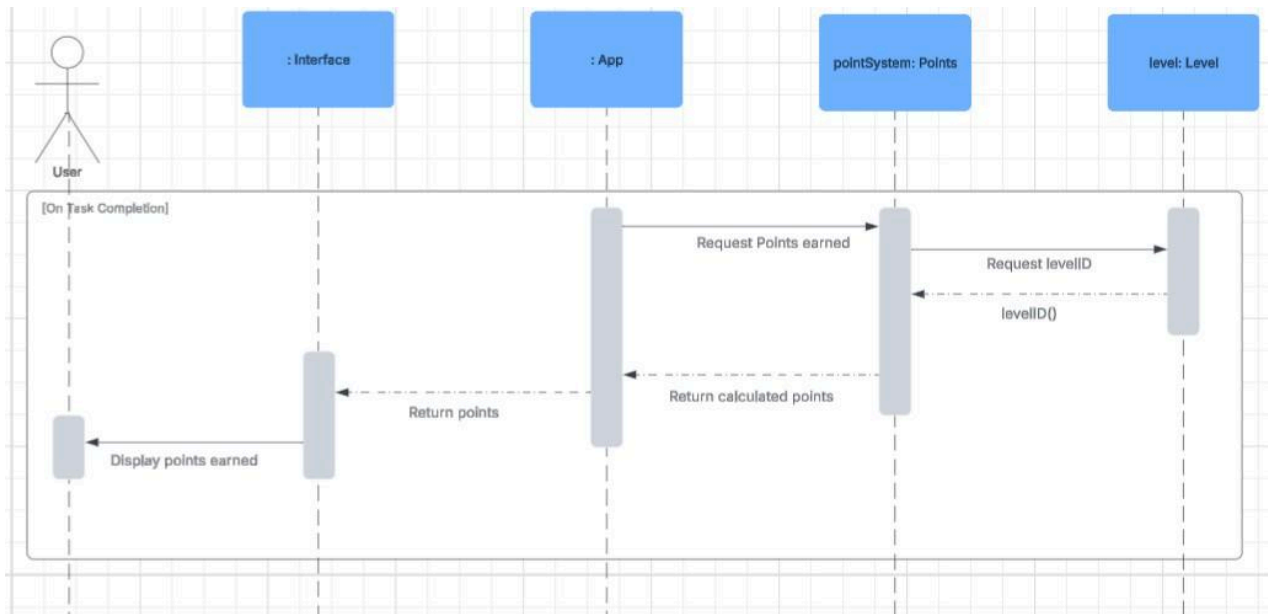### i   Login
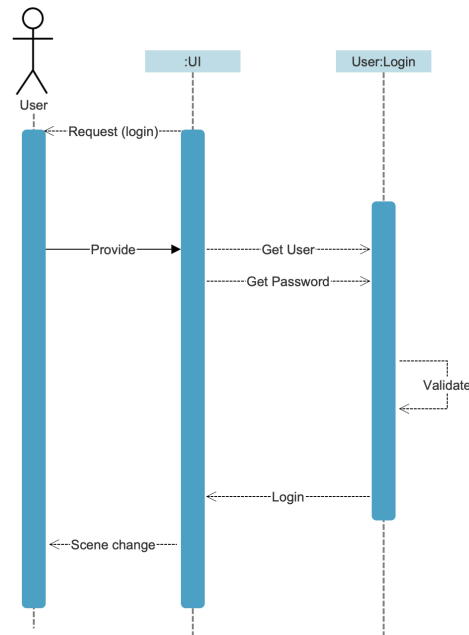
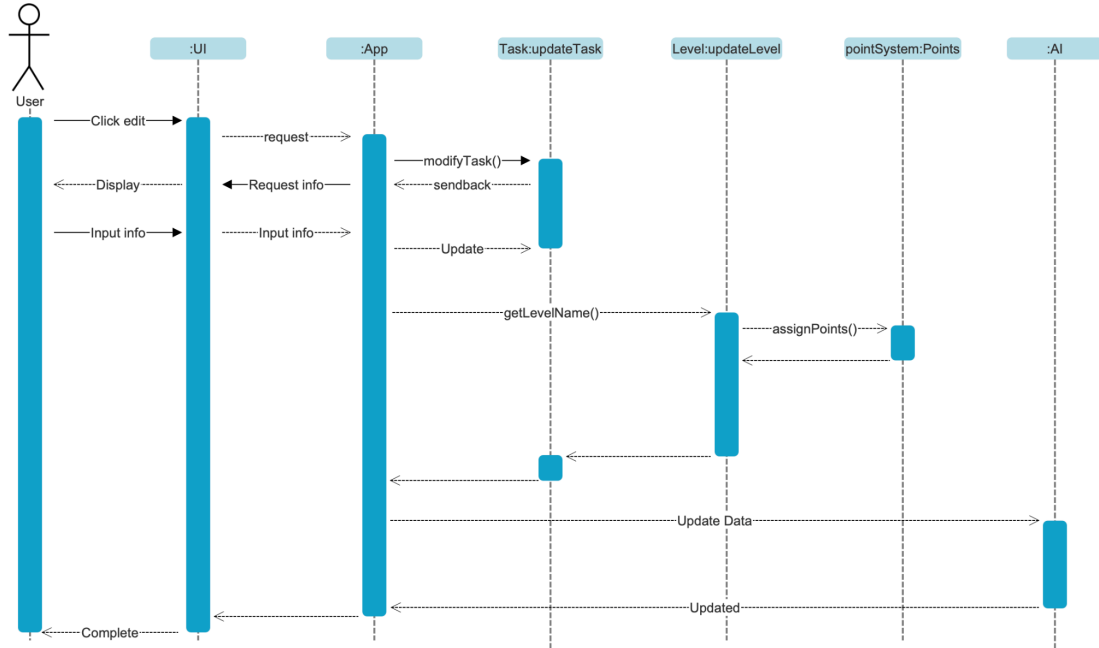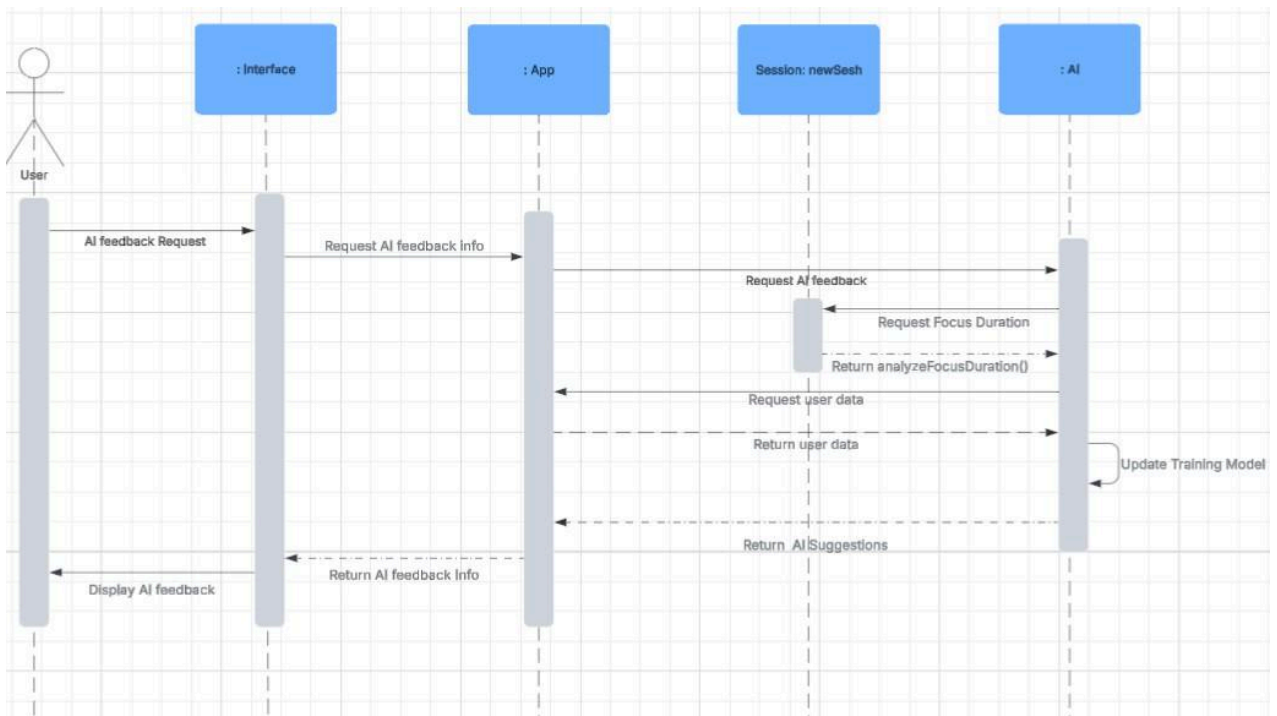### ii   Manage Focus Sessions
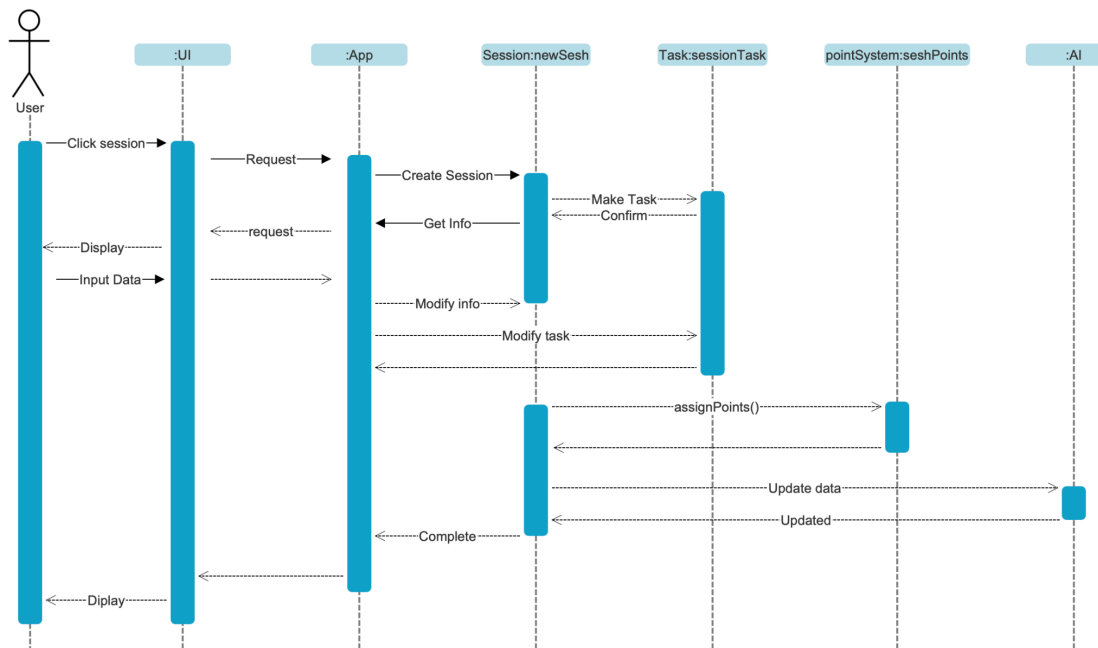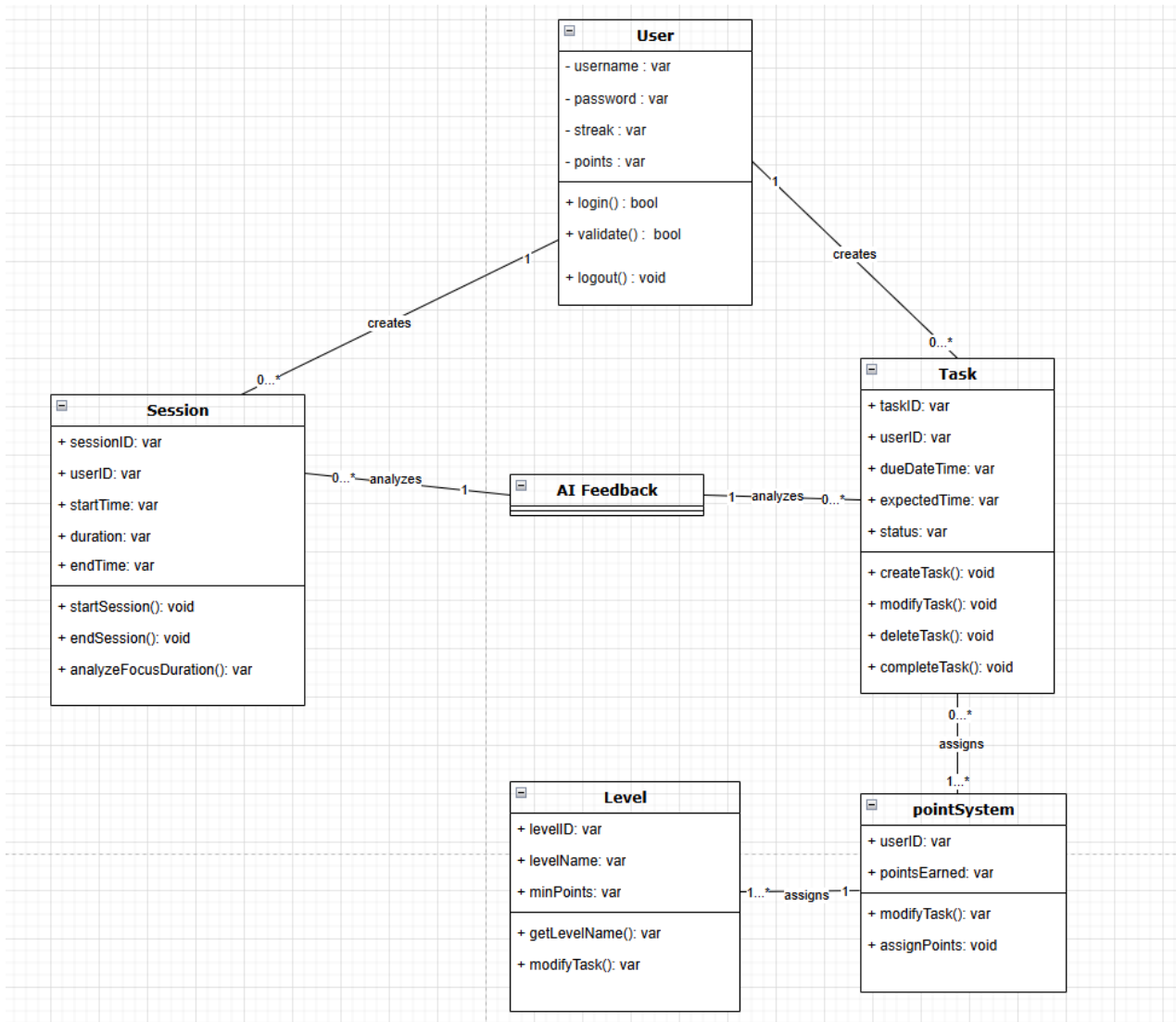
iii   Create Tasks
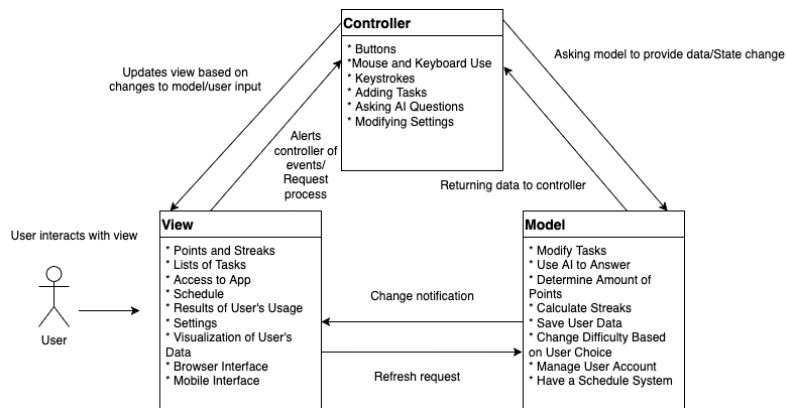

iv   Earn Points

v   Manage Tasks

vi   Ai Feedback

5.4. Class Diagram

User
- username : var
- password : var
- streak : var
- points : var
+ login() : bool
+ validate() : bool
+ logout() : void

creates

creates

1

1

0...*

0...*

Session
+ sessionID: var
+ userID: var
+ startTime: var
+ duration: var
+ endTime: var
+ startSession(): void
+ endSession(): void
+ analyzeFocusDuration(): var

0...*—analyzes—1

AI Feedback

1—analyzes—0...*

Task
+ taskID: var
+ userID: var
+ dueDateTime: var
+ expectedTime: var
+ status: var
+ createTask(): void
+ modifyTask(): void
+ deleteTask(): void
+ completeTask(): void

0...*
assigns
1...*

Level
+ levelID: var
+ levelName: var
+ minPoints: var
+ getLevelName(): var
+ modifyTask(): var

1...*—assigns—1

pointSystem
+ userID: var
+ pointsEarned: var
+ modifyTask(): var
+ assignPoints: void

6. Architectural design

Controller
* Buttons
*Mouse and Keyboard Use
* Keystrokes
* Adding Tasks
* Asking AI Questions
* Modifying Settings

Updates view based on changes to model/user input

Asking model to provide data/State change

Alerts controller of events/ Request process

Returning data to controller

View
* Points and Streaks
* Lists of Tasks
* Access to App
* Schedule
* Results of User's Usage
* Settings
* Visualization of User's Data
* Browser Interface
* Mobile Interface

User interacts with view

User

Change notification

Refresh request

Model
* Modify Tasks
* Use AI to Answer
* Determine Amount of Points
* Calculate Streaks
* Save User Data
* Change Difficulty Based on User Choice
* Manage User Account
* Have a Schedule System

6.1. Describe why the pattern is selected

We've selected to use MVC because we believe that it is the architectural pattern most suited to complete our project within the given timeframe. Its separate three tier component approach will increase the speed of our development process by allowing us to work on the visual, controller, and model aspects of our system simultaneously. Furthermore, the ability to create multiple views for our model aligns well with the mobile/web application nature of our project. The controller and model being unaffected by changes to the view is especially beneficial for a productivity application where changes to the view to enhance user experience is not uncommon.

7. Project Duration, Planning, and Scheduling (Juan & Ndubueze)

7.1.   General Schedule:

Planning and Organization - Up until April 21st:
- Task Delegation
- Review and Revise Deliverable 1
- Project Scheduling & Staffing
- Cost, Effort & Pricing Estimation

Development and Testing - Up until April 25th:
- Test Plan
- Comparison with Similar Designs
- Repository Submission

Finalization and Presentation - Up until April 28th:
- Conclusion
- Presentation Slides (with prototype demo)
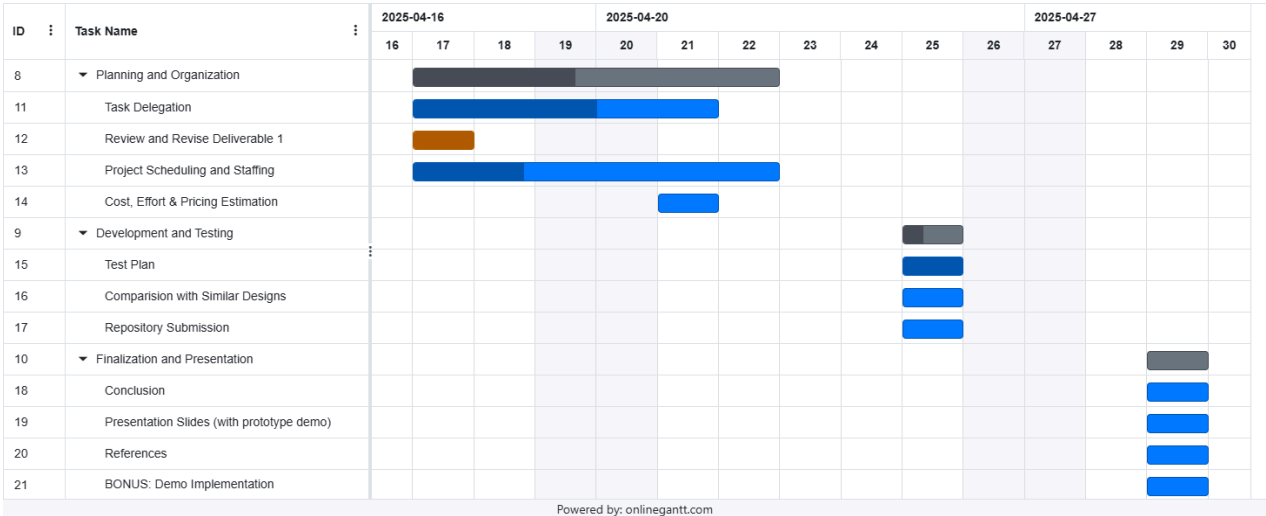- References
- BONUS: Demo Implementation

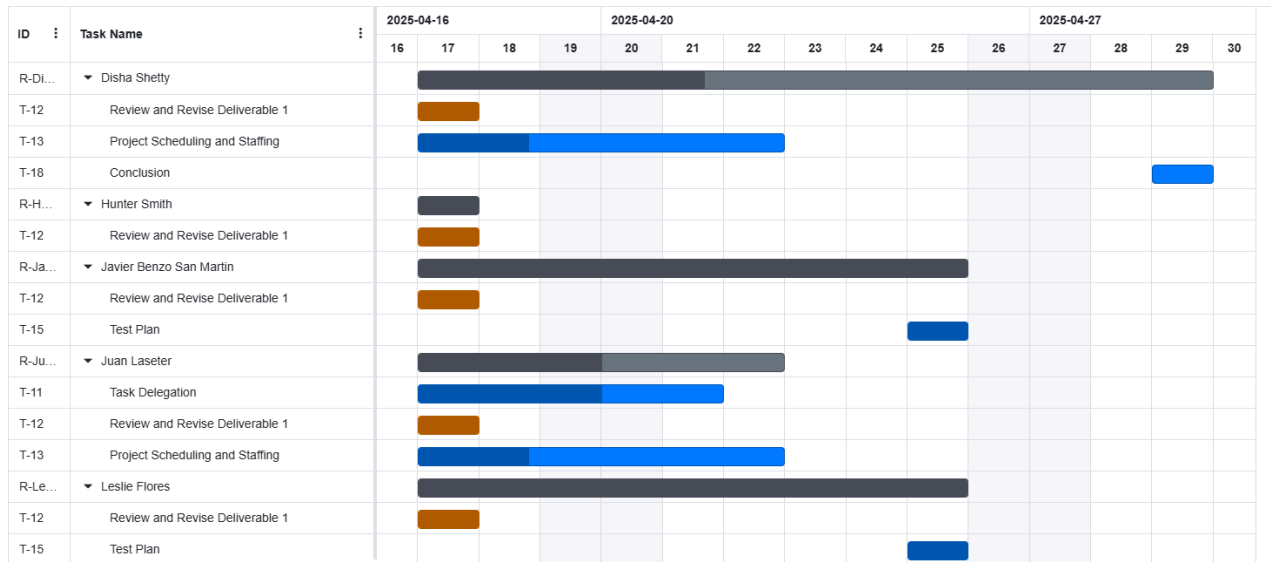| Milestone	Description
M1	Planning Complete (T1 done)
M2	Test Plan Ready
M3	Design Comparison Complete
M4	Estimation Complete
M5	Presentation Content Ready
M6	Review and References Done
M7	Demo Built

M8	Final Deliverables Submitted

Task	Task Name	Effort (person-days)	Duration (days)	Dependencies
T1	Task Delegation	15	10	—
T2	Review and Revise Deliverable 1	4	5	—
T3	Project Scheduling and Staffing	20	15	T1 (M1)
T4	Test Plan	5	10	T1 (M1)
T5	Comparison with Similar Designs	5	10	T2, T4 (M3)
T6	Cost, Effort & Pricing Estimation	10	5	T1, T2 (M4)
T7	Review and Revise Deliverable 1.1	6	5	T1 (M1)
T8	Repository Submission + References	5	5	T4 (M2)
T9	Presentation Slides + Demo	10	15	T3, T6 (M5)
T10	Conclusion	10	10	T7, T8 (M6)
T11	BONUS: Demo Implementation	15	10	T9 (M7)
T12	Final Submission	10	10	T10, T11 (M8)

Gaant Chart

ID	Task Name		2025-04-16				2025-04-20								2025-04-27			
			16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

onlinegantt.com

8. Unit testing (Javier & Leslie)

<u>Login</u>

#	Description	Input	Expected output
1	Successful login	Username= "student1" Password= "corrPass"	Login successful, dashboard open
2	Failed login - Wrong password	Username= "student1"	Error: incorrect password

		Password= "wrongPass"	
3	Failed login - Non-existing user	Username= "unknown" Password= "anyPass"	Error: user not found
4	Faied login - Empty fields	Username= "" Password= ""	Error: Missing username or password.
5	Create account successfully	Username= "newUser" Password= "newPass"	Account created, dashboard opens
6	Create account failure	Username= "student1" Password= "anotherPass"	Error: Username already exists.

Manage Focus Sessions

#	Description	Input	Expected Output
1	Start a focus session normally	Press "Start Session" button	Session timer starts
2	End a focus session after 45 min	Press "End Session" button	Session saves with 45 min duration
3	Start session without ending previous	Press "Start Session" while session active	Error: session already active
4	Session ends due to app crash	App crashes mid-session	Session auto-saved up to last active minute

Create tasks

#	Description	Input	Output
1	Create task with valid inputs	Title= "Finish assignment" Due= "05-01-2025"	Task created successfully
2	Create task without title	Title= "" Due= "05-01-2025"	Error: Task title required
3	Create task with past due date	Title= "Old task" Due= "01-01-2025"	Error: Due date must be in future
4	Create duplicate task	Title= "Finish assignment" Due= "05-01-2025"	Error: Task already exists

Earn Points

#	DESCRIPTION	INPUT	EXPECTED OUTPUT
1	Earn points after task completion	User completes a task with 30 min expected time	User earns 30 points
2	Earn points after multiple tasks	User completes three tasks (10-, 15-, and 20-min tasks)	User earns 45 points (sum of all expected times)

3	Earn points for over task	User completes a task past its due date	User still earns points based on expected time

Manage Tasks

#	DESCRIPTION	INPUT	EXPECTED OUTPUT
1	Edit a task's due date	User changes due dat of task	Task is updated with new due date
2	Edit a task's expect time	User updates expecte time of task to 120 minutes	Task expected time is updated to 120 minutes
3	Delete a task	User deletes task "Task1"	"Task 1" is removed from the task list

AI Feedback

#	DESCRIPTION	INPUT	EXPECTED OUTPUT
1	Get encouragement after task completion	User completes a focus session	AI chatbot sends motivational message
2	Get personalized tip based on past sessions	User requests focus ti after completed sessions	AI chatbot gives a personalized tip related to focus trends
3	Get peak productivity time suggestion	User asks for productivity analysis after multiple sessions	AI suggests best times based on historical focus data

9. Similar Design Comparisons

The Habitica Application also provides a similar gamified approach towards task productivity. The application converts tasks and goals into a game where the user "levels up" based on the user's level of productivity. Habitica also provides in-game consequences for missing goals or engaging in bad habits, further enhancing the productivity-boosting element of the game. This is what makes Habitica unique; The utilization of behavioral psychology by using positive reinforcement to create habit loops.

8.1.   The main features of this application are:
i   Task Categories : Habits, Dailies, and To-Dos
ii   Gamified Progression : Experience Points and Gold gains when completing a task, which can be put towards buying in-game rewards.
iii   Damage and Penalties : Missing tasks or bad habits results in taking damage and can lead to loss of in-game rewards or items

      The Adventurefy application takes the gamified approach towards task productivity by converting the user's to-do list into a series of quests, where completing real-life tasks helps the user progress in the game. The user's in-game character has abilities which can be further enhanced by experience points.

8.2.   The main features of this application are:
i   Habit Tracking : track current or recurring behaviors that the user may want to build-up or reduce
ii   To-Dos: Manage one-time tasks that contribute to overall progress
iii   Gamified Progression : Earn Experience points that allow the user to enhance their in-game character

10.1 Estimated Cost of Hardware

Item	Description	Quantity	Unit Cost	Total Cost	Notes
Laptop	Needed for Software Development Such as Coding, Testing, and Design	9	$1,400	$12,600	One per Team Member
Monitor	External Monitor for Better Productivity	9	$300	$2,700	Optional but Will Be Provided Multitasking as Its Helpful
Keyboard	Improves Typing Comfort	9	$50	$450	Optional Setup
Mouse	Improves Comfort and Can Be Helpful During Design and Development Tasks	9	$30	$270	Optional Setup
Headphones	For Meetings to Improve Focus	9	$100	$900	Optional Setup
Webcam	For Remote Collaboration and Meetings	9	$80	$720	Required for Team Meetings
Microphone	Better Audio Quality for Team Presentations	9	$100	$900	Required for Team Meetings
			$2,060	$18,540	

10.2 Estimated Cost of Software Products

Category	Software Product	Purpose	License Type	Cost per License	Quantity Needed	Billing Cycle	Total Estimated Cost
AI and Machine Learning Services	Google Gemini API	AI Functionality Integration	Paid	Depends on Usage	N/A	Pay-As-You-Go	$6,615
Cloud Services and Hosting	AWS	Backend Hosting	Paid	Depends on Usage	N/A	Pay-As-You-Go	$1,800
Design and Prototyping Tools	Figma Professional	UI/UX Designing	Paid	$20	9	Monthly	$180
Development Tools (IDEs and Utilities)	All Products Pack	Full Coding Environment Suite	Paid	$77.90	9	Monthly	$701.10
Monitoring and Analytics	Google Analytics (GA4)	App Monitoring/User Metrics	Free	N/A	N/A	N/A	N/A
Office and Documentation Suites	Google Workspace Business	Docs, Email, Drive, etc.	Paid	$14	9	Monthly	$126
Platform Access and Developer Programs	Apple Developer Program	App Store Distribution Access	Paid	$99	1	Annual	$99
Project Management and Productivity	Trello	Team Management	Paid	$10	9	Monthly	$90
Version Control and Collaboration	GitHub Team Plan	Code Repository and Collaboration	Paid	$4	9	Monthly	$36
				$225			$9,647

10.3 Estimated Cost of Personnel

#	Role	Experience Level	Hourly Rate ($)	Weekly Cost ($)	Estimated Duration (in Weeks)	Total Cost
1	Project Manager/Project Analyst	Mid-Level	$ 55.29	$ 2,211.60	16	$ 35,385.60
2	Front End Developer	Mid-Level	$ 48.08	$ 1,923.20	16	$ 30,771.20
3	Front End Developer	Junior	$ 33.14	$ 1,325.60	16	$ 21,209.60
4	Backend Developer	Mid-Level	$ 52.88	$ 2,115.20	16	$ 33,843.20
5	Backend Developer	Junior	$ 38.46	$ 1,538.40	16	$ 24,614.40
6	AI/ML Engineer	Mid-Level	$ 69.17	$ 2,788.40	16	$ 44,614.40
7	AI/ML Engineer	Junior	$ 39.85	$ 1,594.00	16	$ 25,504.00
8	UI/UX Designer	Junior	$ 39.74	$ 1,578.80	16	$ 25,260.80
9	Quality Assurance Engineer	Junior	$ 35.96	$ 1,438.40	16	$ 23,014.40
			$ 412.57	$ 16,513.60		$ 264,217.60

11. Function Point

Functionality	Details	Simple	Average	Complex	Complexity
User Inputs	- Variable input, dynamic based on UI fields (`ai-page.tsx`). - Worst case: 10 - Supports multiple messages and interactive elements.	3	5	6	50
User Outputs	- Dynamic output generated by AI in response to inputs. - Worst case: 5 - Messages are appended sequentially to the UI.	4	5	7	20
User Queries	- Worst case: 8 primary queries handled: - Additional endpoints for user data and focus sessions.	6	8	9	64
Data Files	Worst case: 30 files	6	7	10	300
Relational Tables	- 3 Key tables: • `users` • `tasks` • `focus_sessions`	3	6	8	24
External Interfaces	- 4	3	6	10	40

GFP: 498

PCA Computation

PC = 0(no influence) 1(incidental) 2(Moderate) 3(Average) 4(Significant) 5 (essential)

(1) Does the system require reliable backup and recovery? 1

(2) Are data communications required? 5

(3) Are there distributed processing functions? 2

(4) Is performance critical? 2

(5) Will the system run in an existing, heavily utilized operational environment? 3

(6) Does the system require online data entry? 5

(7) Does the online data entry require the input transaction to be built over multiple screens or operations? 4

(8) Are the master files updated online? 5

(9) Are the inputs, outputs, files, or inquiries complex? 4

(10) Is the internal processing complex? 4

(11) Is the code designed to be reusable? 3

(12) Are conversion and installation included in the design?3

(13) Is the system designed for multiple installations in different organizations? 0

(14) Is the application designed to facilitate change and ease of use by the user? 5

Processing Complexity Adjustment (PCA): PCA = 0.65 + 0.01 * PC

PC = 46.

PCA = 0.65 + 0.01 *46 = 1.11

FP = GCP * PCA

498*1.11 = 552.78

Productivity = FP/ (no of people* no of weeks) = 552.78/ 10 *9 = 6.14 FP/person-week

E = FP/productivity = 552.78/6.14 = 90.03

In conclusion, since our team size = 9, then project duration is: D = E / team size = 90.03/9 = 10.003 (round up to 11)

12. Challenges and Changes in Project Management, in Software Planning etc.

During the software planning phase, we encountered a broad range of potential implementation ideas. However, considering the restraints of our timeline and currently available resources, we made the decision to prioritize the most critical features for the first release. Additional functionalities are to be scheduled for future versions to allow for a phased rollout without compromising the quality of our core product.

A key challenge we experienced was distinguishing our application in an already mature productivity app market. With so many options, thinking about what makes FocusQuest unique was important. To address this we conducted market research early in the planning phase to define a clear, unique value preposition. Our deliberate design and functionality decisions ensured that the product would meet user needs in a unique way.

As we used waterfall methodology, detailed upfront planning was needed. Once requirements were finalized we kept changes to a minimum to maintain schedule and avoid rework. Through careful documentation at each stage, we ensured alignment among all stakeholders.

By maintaining clear communication, following a disciplined project structure, and carefully balancing ambition and feasibility, we were able to successfully navigate the challenges in software development and planning.

13. Prototype
Link:
https://www.figma.com/design/FyQoafpJh62UBr2Vc5HFM6/Prototype?node-id=39-4138&t=eIs3CtAeiGQEscHW-1
Figma Prototype password: savant-lock-wool-honey

14. References
Tools:

- "Replit." *Replit*, replit.com. Accessed 15 Apr. 2025.

- "Figma." *Figma*, figma.com. Accessed 13 Apr. 2025.

API:

- Google. "Gemini API Documentation." *Google AI*, 2024, ai.google.dev.

Similar Apps:

- "Adventurefy." *GitHub*, github.com/Adventurefy. Accessed 29 Apr. 2025.
- HabitRPG. "HABITRPG/Habitica: A Habit Tracker App Which Treats Your Goals like a Role Playing Game." *GitHub*, github.com/HabitRPG/habitica. Accessed 29 Apr. 2025.