Project Report:
Sentence Squabble

A game where the player must use the provided words in a sentence of their making. No matter how sillly the result.

CITA 204 (0W1) Group 2:
Matt Begnoche,
Jacob Ramos,
Abby Rodenas,
Jonathan Zimmermann

December 5, 2023

Table of Contents

1 Exectutive Summary

Sentence Squabble is a word game that was specifically developed for android phones. Our game generates five words which the user must use to create a grammatically correct sentence. We developed Sentence Squabble using Android Studio deciding to split our team into two groups. Group one developed the “backbone” of the game while the other developed the UI and layouts.

1.1 Project Benefits

- A fun and engaging new game for users to enjoy.
- All players are given equal opportunity (receive the same 5 words) to complete the daily challenge.
- The brain benefits from a more mentally stimulating experience.
- Users can work on their sentences throughout the day by saving a draft.
- A new set of words will generate daily to encourage users to continue using the app.
- A practice mode will allow users unlimited practice rounds to master their sentence making skills.
- Users can share their sentences with friends and compare results, encouraging others to use and continue using the app.
- A user-friendly experience and easy to navigate application.
- User encouragement with positive feedback from the app upon completion of a grammatically correct, 5-word sentence.
- Low-cost maintenance.
- Opportunity for users to provide feedback via surveys.

1.2 Product Objectives

1. The application presents the player with five words which contain five letters, with the generated words the player must create a sentence that is grammatically correct.
    - Create a field where the player can enter their sentence.
    - Create a method that can detect when the player has finished typing their sentence.
2. Once the player has entered their sentence the application will use a process to detect whether the sentence is grammatically correct and contains all the generated words.
    - If both requirements are met the player is greeted with a winning pop up.
    - If the player wins, they are prompted with an option to share their sentence digitally.
3. Develop a welcome screen that allows the user to choose between a “daily challenge” mode or a “practice” mode.
    - All players must be presented with the same five daily words.

- Practice mode cannot contain the words presented in the daily challenge.

4. Create a way for the player's progress to be saved.
5. Implement advertisements to create a path for monetization.
6. Develop a UI that is clean and simple to understand.

1.3 Product Objectives Fullfillment Strategy

Each day a new set of 5, 5-letter words will be generated for users based on a random seed that ensures each user has the same opportunity to complete a sentence using the same words, no matter where they are located. We've created a field within the daily challenge mode for the user to input their sentence, as well as a button for users to acknowledge that they have completed their sentence and is ready for grammatical verification. Once the user submits their daily challenge (or practice mode sentence) the application references a grammer checker to verify that the sentence follows proper grammar practices. If the sentence meets the requirements of being grammatically correct and containing all given words, users are greeted with a screen acknowledging they are a winner (otherwise a screen stating that they are not a winner this round is presented). Utilizing Android Studio and Java, the application was designed to have both a daily challenge mode that allows users to save their progress and return later, a practice mode, a main welcome screen, and lastly a screen to share results externally from the app all with a simple and clean UI. Within the practice mode, the application ensures users are not receiving any of the same words they receive within the daily challenge mode in order to preserve the integrity of daily challenge.

1.4 Verification and Validation

Through survey data and subjective testing, it was determined that Sentence Squabble would be well received by audiences and performs similarly to similar applications. Through play testing additional improvements were identified to target to create a viable commercial product.

2 Development Approach

The core functionality of the game was created by the Systems Analysts team in Android Studio, leveraging GitHub for version control and change management to facilitate collaboration. Grammar checking and the dictionary were sourced from third parties. LaTeX was used to create documentation and OneDrive as the document store to facilitate further collaboration. Meetings were held weekly to promote project progression and ensure project creep is minimal. Tasks and resources were managed using LibreOffice as well as to verify dependencies and requirements are met within the allocated time-frame while managing resource usage to ensure optimal utilization.

2.1 Planning

Planning for the prototype development of Sentence Squabble took place from August 24th, 2023, until September 28th, 2023. During the planning stage the team created a Statement of Purpose and Project Proposal outlining the benefits, requirements, objectives, projected schedule, and cost of the prototype development. The initial schedule split the prototype development into three main stages, Prototype Skeleton Development, Prototype Development, and Prototype Verification and Validation which would be followed up by the creation of this report. All work was predicted to be completed by December 5th, 2023.

2.2 Prototype Skeleton Development

The Prototype Skeleton Development stage consisted of setting up the project in Android Studio and creating the code base that would form the title page, practice mode, and challenge mode for the Sentence Squabble application as well as the transitions between them. This also established the interface methods between the different portions of the application. This portion of the development went smoothly and was completed ahead of schedule on October 3rd, 2023.

2.3 Prototype Development

The Prototype development stage of the development consisted of splitting the team into two separate groups, one group would work on the application functionality, and the other would work on the layout and visuals. The functional development had several hurdles to properly integrate LanguageTool for grammar checking and Wiktionary for sourcing the 5, 5 letter words into the application. However, these development challenges were overcome, however would require additional refinement for a successful commercial product. The visual development was successful and produced a polished and clean look for the application. Due to several setbacks' completion of the Prototype Development stage was nearly two weeks later than predicted ending on November 10th, 2023.

2.4 Prototype Verification and Validation

The Prototype Verification and Validation consisted mostly of collecting surveys from play testers and a subjective analysis of the application's performance when compared to similar word game applications. Due to the delay in the Prototype Development stage the Prototype Verification and Validation stage started later than anticipated. However, 10 play testers were given the application and the survey and Sentence Squabble received overall positive feedback. The Prototype Verification and Validation stage concluded on December 1st, 2023, approximately two and a half weeks after the original prediction. This delay was mostly due to the large amount of time required to get reliable play testers and gather feedback. However, the original project deadline of December 4th, 2023, was only missed by one day ending on December 5th, 2023 which was still ahead of the project due date of December 8th, 2023.

3 Systems Analysis

3.1 Constraints

A number of constraints were identified and listed in Table 1. The most cricital constraint for the project was the limited allocation of time. This was followed by the social and contextual word constraints due to the prototype nature of the game and limited time to refine the grammer checking algorithm.

Table 1: Constraint Details

Constraint	Type	Timing	Urgency	Notes
Remote Collaboration	Internal	Present	Mandatory	Remote Collaboration presents unique challenges and benefits.
Limited Allocation of Time	Internal	Present	Mandatory	Due to work or school, project members each have limited time to allocate towards the project.
Social Interaction	External	Present	Desirable	The game must have an optional social concept (eg. sharing, communicating, competing).
Contextual Word Selection	External	Present	Mandatory	The selected words for any problem presented to the user must not be impossible.
Daily Challenge must be the same for all users	External	Present	Mandatory	If the Daily Challenge differed then players could not share and compare results, damaging the integrity of the social aspect of the game.
Android	Internal	Present	Desirable	The game will be created for Android.
On-Device Storage Only	External	Future	Desirable	On device storage is preferable, therefore, user data should be kept at a minimum. Random seeds will need to be formulaic to allow reproduction on every device. Player discovery of the fomula could damage the game's integrity.
Unique Practice Mode Words	External	Future	Mandatory	If users were able to practice the daily challenge words, it would damage the integrity of the competitive nature of the Daily Challenge.
Keep user personal data secure	External	Future	Mandatory	Exposing personal data would be damaging for the player and ruin the reputation of the game. It is important to keep used personal data secure.
Use minimual personal data	External	Future	Mandatory	Exposing personal data would be damaging for the player and ruin the reputation of the game. Minimal personal data should be used to ensure the security of such data.

3.2 Product Specification Fullfillment

The product specifications in Table 2 were derived from the project objectives using requirements decomposition using a LibreOffice Calc spreadsheet and did not change through the course of the development.

Table 2: List of Product Specifications

ID	Title	Specification	Obj. ID(s)	Fullfillment
S001	Number of Words Presented	5	1	The player was presented with 5 words.
S002	Number of Letters in Words Presented	5	1	Each of the 5 words the player was presented consisted of 5 letters.
S003	Challenge Mode	Once a day players will be presented the number of unique words specified in S001 to create their sentences using.	3	The challenge mode consisted of the player being presented 5 unique 5 letter words daily to use while creating their sentence.
S004	Save Sentence Under Construction	The sentence currently being constructed in a given game mode shall be saved.	2	For both the practice and challange modes the sentence under construction was saved.
S005	Save Player Usage Frequency	The player usage frequency will be saved. Each time a sentence is evaluated will count as 1 “usage”.	4	The number of sentence evaluations were recorded by the application.
S006	Save Player Challenge Shares	The number of times the player shares their winning challenge sentence will be saved.	4	The number of player win shares was recorded by the application.
S007	Save Challenges Player Does Not Share	The number of times the player does not share their winning challenge sentence will be saved.	4	The number of times the player does not share their win was recorded by the application.
S008	Save Player Challenge Wins	The number of times the player wins challenge sentence will be saved.	4	The number of times the player won was recorded by the application.
S009	Save Player Challenge Looses	The number of times the player submits their sentence for the challenge but does not win will be saved.	4	The number of times the player submits their sentence and looses was recorded by the application.
S010	Dictionary of English Words	The dictionary of English words shall be derived from the open source at the following link: https://github.com/dwyl/english-words.	1, 2	The words were derived from Wiktionary instead. Still meets the dictionary of english words requirement.
S011	Only Store Usable Words	The application shall only contain words that will be used by the application. See S002 for the number of letter words.	1	Only the words required by the application were downloaded to the application.
S012	Grammar Checking	Grammar checking shall be done using the dev.languagetool.org Java library.	2	Grammer checking was implemented using languagetools.
S013	Advertisement Style	Banner Ads shall be used for advertisement presentation.	5, 6	Banner Ads were implemented in the game mode screens.
S014	Advertisement Service	Google Mobile Ads shall be used to serve advertisements.	5	Google Mobile Ads was used to serve the Ads.
S015	Title Screen for Game Mode Switching	There shall be a title screen to facilitate switching between game modes.	3, 6	The title screen did facilitate switching between game modes.
S016	Daily Challenge Reset	The daily challenge shall reset at midnight according to the phone’s local time.	5	The daily challenge does reset at midnight local time.
S017	Sentence Evaluation Button	The evaluation input shall be a button.	2, 6	For both the practice and challange modes the sentence evaluation input was implemented as a button.
S018	Return to Title Screen Button	The return to title screen input shall be a button.	3, 6	For both the practice and challange modes the return to title screen input was implemented as a button.
S019	Player Sentence Text Box	The player shall enter their sentence into a text box.	2, 6	The player was provided a box to input their sentence.

3.3 Context Diagram

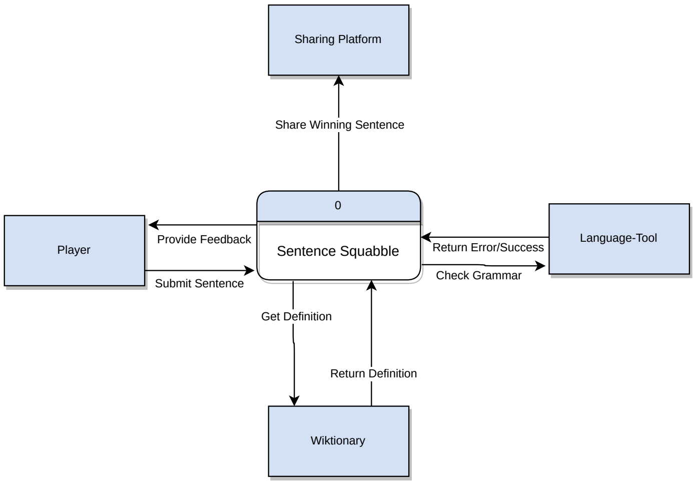


Figure 1: Sentence Squabble Context Diagram

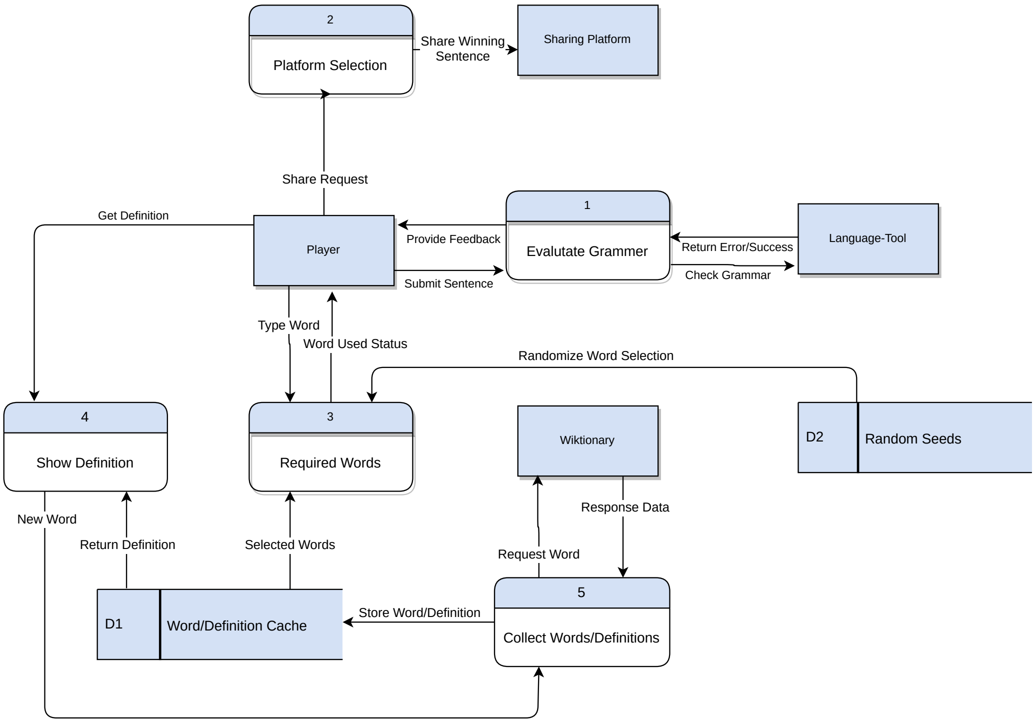


Figure 2: Sentence Squabble Context Diagram 0

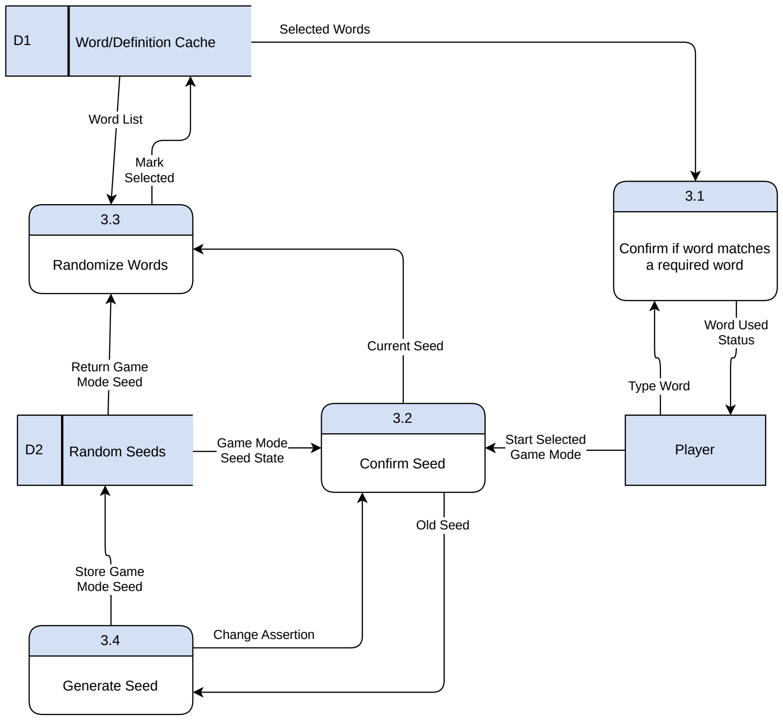


Figure 3: Sentence Squabble Context Diagram 3

3.4 Dataflow Diagrams

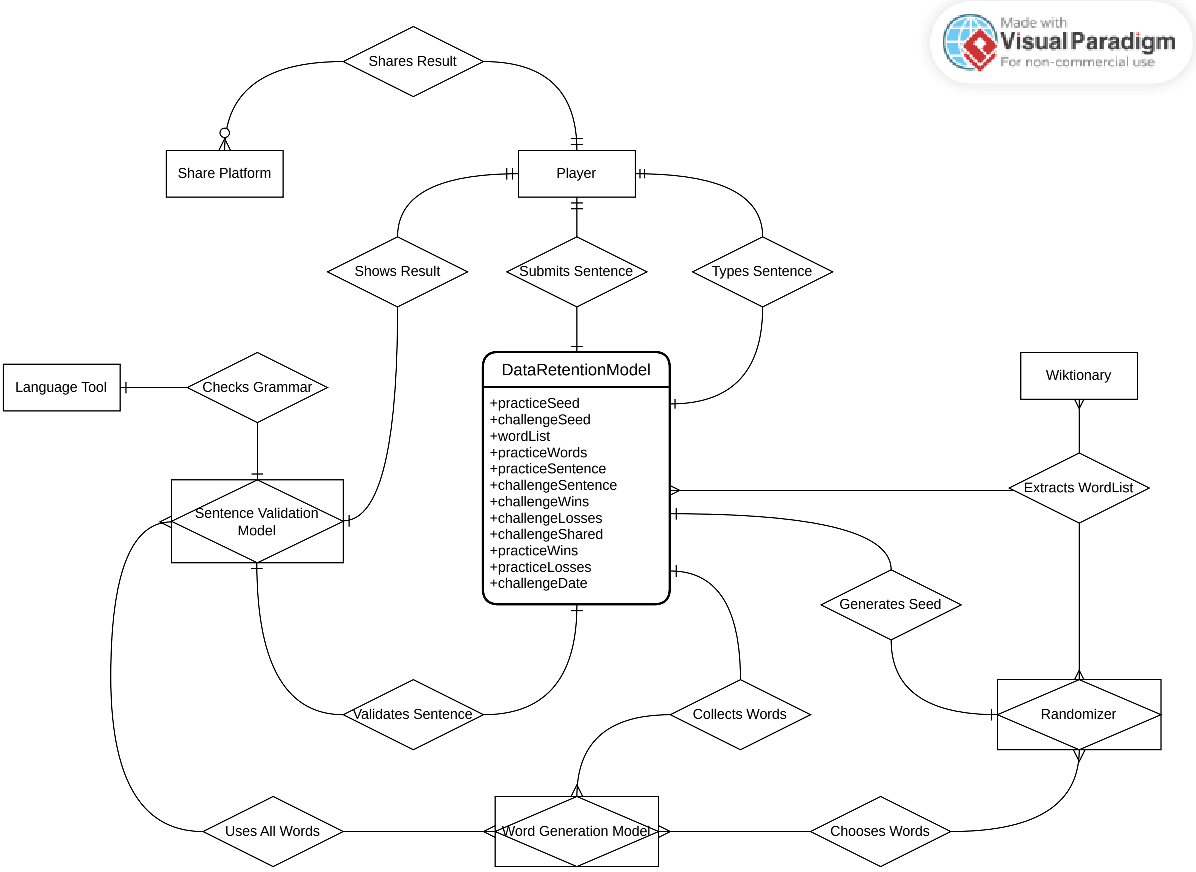


Figure 4: Sentence Squabble E-R Diagram

3.5 System Architecture Diagram

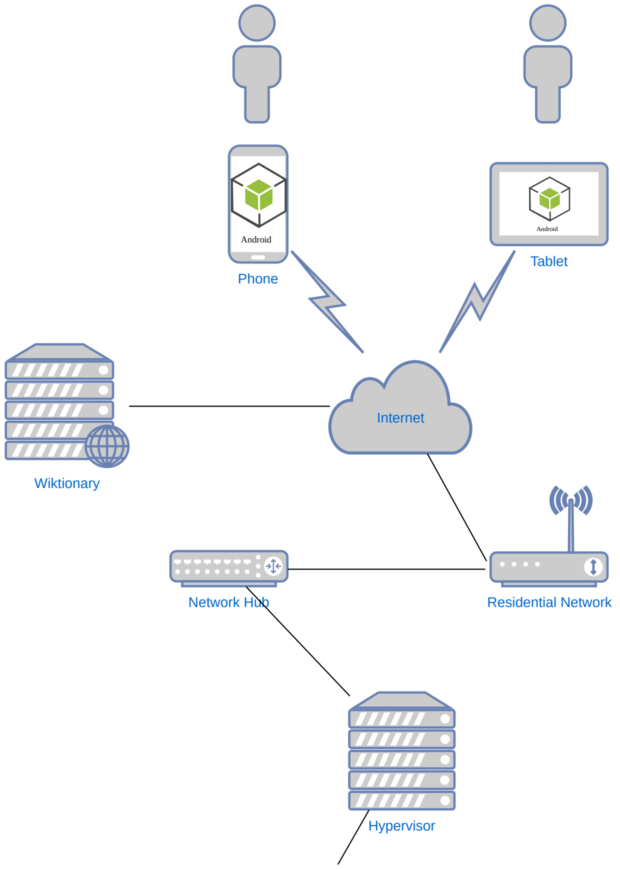


Figure 5: Sentence Squabble System Architecture

3.6 Prototype Development Costs

The total cost for prototype development was estiimated to be $74,777.25, however ended up costing $11,949.18 due to the additional amount of developer time required to create the prototype application. The total cost was calculated assuming development began with the creation of the project proposal on September 6th, 2023 and concluded when the final report was due on December 8th, 2023. Using Glassdoor as a reference, a $120,000 salary was assumed for each of the four systems analysts/core developers that comprise our team. The cost breakdown in Table 3 shows that the primary development cost is personnel followed up by hardware, while the majority of the software and tools being utilized for the project are either open source or free.

Table 3: Prototype Development Cost Breakdown

Resource	Type	Schedule	Est. Cost	Act. Cost
ProjectLibre	Productivity	N/A - Free	$0	$0
LaTex	Utility	N/A - Free	$0	$0
LibreOffice	Productivity	N/A - Free	$0	$0
Discord	Collaborative	N/A - Free	$0	$0
Google AdMob	Service	N/A - Free	$0	$0
Personal Computer	Hardware	One-Time	$4,800.00	$4,800.00
Power	Operational	Yearly	$34.42	$76.55
Time	Personnel	Weekly	$5,372.66	$11,949.18
GitHub Basic	Collaborative	N/A - Free	$0	$0
Android Studio	Development	N/A - Free	$0	$0
Language Tool	Library	N/A - Free	$0	$0
english-words	External API	N/A - Free	$0	$0
Xcode	Development	N/A - Free	$0	$0
Google Player Developer Console	Platform	One-Time	$25.00	$25.00
Apple Developer Program	Platform	Yearly	$99.00	$99.00

The commercial product development cost projection in Table 4 was conducted based upon the prototype development, and additional required costs and tools. The projection was done assuming that commercial development would start on January 1st, 2024, and end July 14th,2024. Additional software packages and developer tools would be required to ensure the application could be properly distributed to all platforms and phones. This projection includes the Google Player Developer Console and the Apple Developer Program costs since they were not actually paid to develop the prototype. The total cost to make Sentence Squabble into a commercial product was projected to be $406,474.54.

Table 4: Prototype Development Cost Breakdown

Resource	Type	Schedule	Est. Cost
GitHub Team	Productivity	Monthly/User	$3.67
Power	Operational	Yearly	$200.14
Time	Personnel	Weekly	$15,620.60
Google Player Developer Console	Platform	One-Time	$25.00
Apple Developer Program	Platform	Yearly	$99.00

4 Prototype Design: Sentence Squabble Android Application

4.1 Title Menu

Upon starting the application, the title page will be displayed, as shown in Figure 6. The layout is simple, giving the player two game play options: the Daily Challenge or Practice Mode.

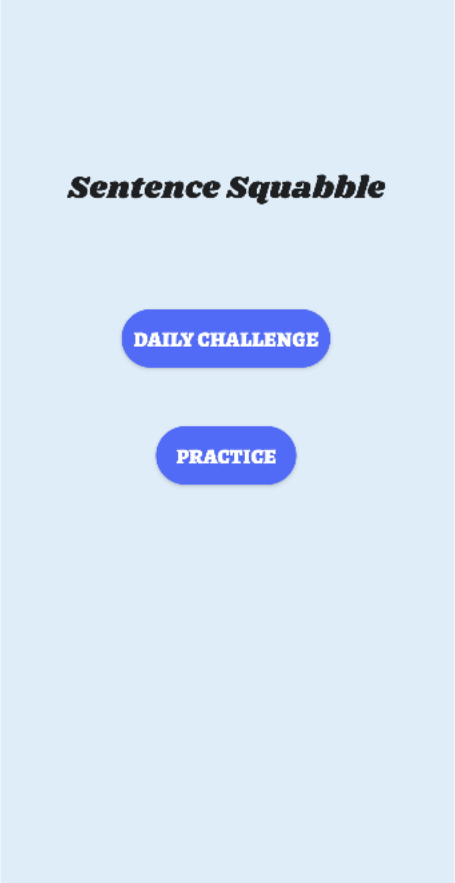


Figure 6: Prototype Design - Title Menu

4.2 Challenge Mode

Upon clicking the Daily Challenge button on the title page, the user will be brought to the Daily Challenge page. The Daily Challenge page, Figure 7, provides the player the 5 daily challenge words to be used to complete a grammatically correct sentence. There is a field for the player to type in their sentence and while the user types, if a daily challenge word has been included in the sentence, the app will acknowledge this by changing the background color of the word's bubble to green as shown in Figure 8.

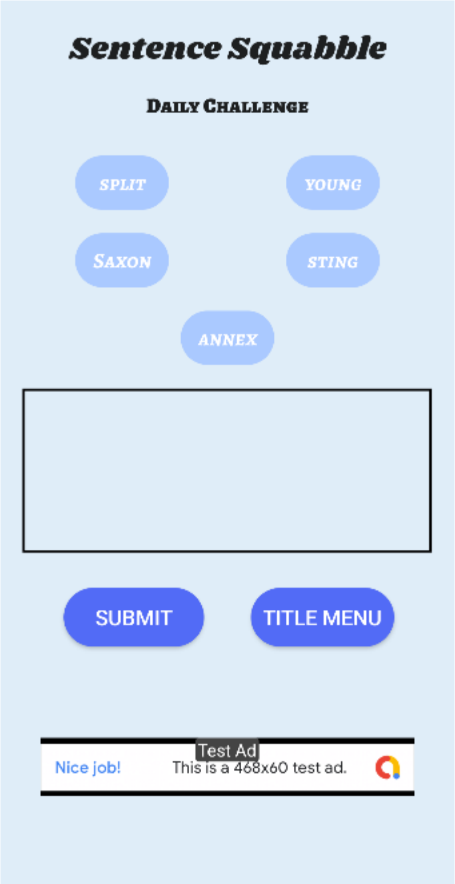
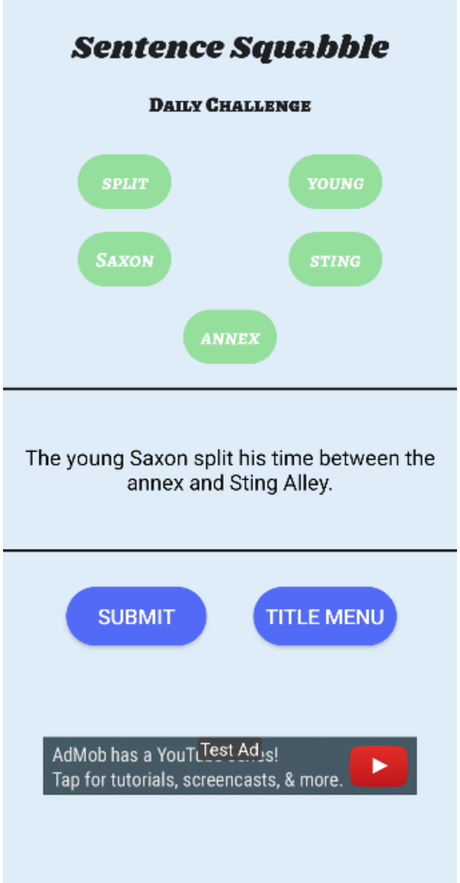


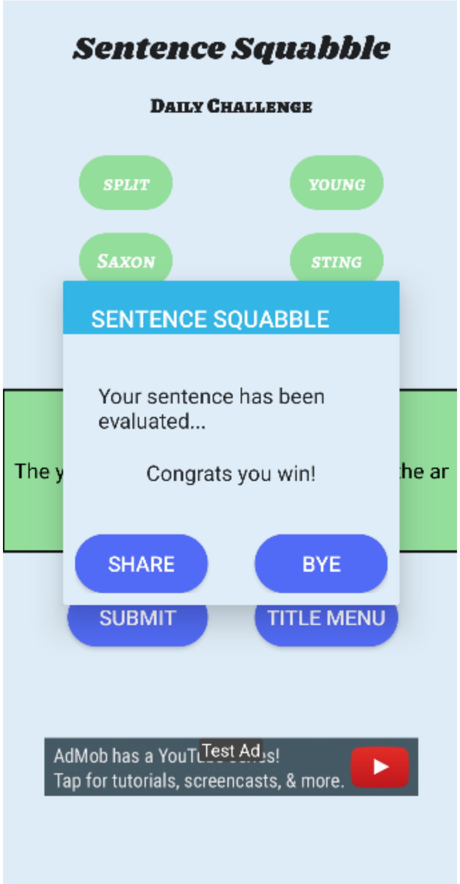
Figure 7: Prototype Design - Daily Challenge Page



The young Saxon split his time between the annex and Sting Alley.

Figure 8: Prototype Design - Daily Challenge Word Detection

At any point in time the player is able to click the “Title Menu” button to go back to the Title Page or close the application and their thus far created sentence will be saved and remain in the field for them to continue working on it. Once the player has decided they would like to submit their daily challenge sentence for verification, they will click the submit button. If the sentence passes grammatical verification and contains all 5 daily challenge words, the player will be presented with the Daily Challenge Win Pop-Up exemplified in Figure 9.The Win Pop-Up notifies the player that their sentence has been evaluated and they have won the day’s daily challenge. From here, the player can either dismiss this message, which will bring them back to the Daily Challenge Page from which they can exit back to the Title Page, or the option to share their sentence with family or friends and challenge them to complete the Sentence Squabble Daily Challenge for themselves. The “Share” button activates the Android Share Module, allowing players to share via their method of choice - text, email, social media, etc. Let it be noted that Google ads have been integrated in a seemingly non-intrusive manner to the Daily Challenge Page.



Your sentence has been evaluated...

Congrats you win!

AdMob has a YouT[illegible]s!
Tap for tutorials, screencasts, & more.

Figure 9: Prototype Design - Daily Challenge Win Pop-Up

4.3 Practice Mode

Upon clicking the “Practice” button on the Title Page, the player will be brought to the Practice Mode shown in Figure 10. Highly similar to the Daily Challenge Page, the Practice Mode presents player with 5 random words to be used to complete a grammatically correct sentence. The practice mode words will never be the same as any of the 5 Daily Challenge words and will allow the player to reset the words as many times as they wish to continue practicing building sentences. There is a field for the player to type in their sentence and while the player types, if a practice word has been included in the sentence, the application will acknowledge this to the player and the word bubble will change to green. If the player would like to verify their sentence is grammatically correct, they can click the “Submit” button. If the sentence passes verification, a small dialog stating “You Win!” is displayed on the bottom of the screen and the text field changes to green. The player can click the “Reset” button as many times as they wish to generate 5 new, random words or click the “Title Menu” button to return to the Title Page. Let it be noted that Google ads have been integrated in a seemingly non-intrusive manner to the Practice Mode Page.

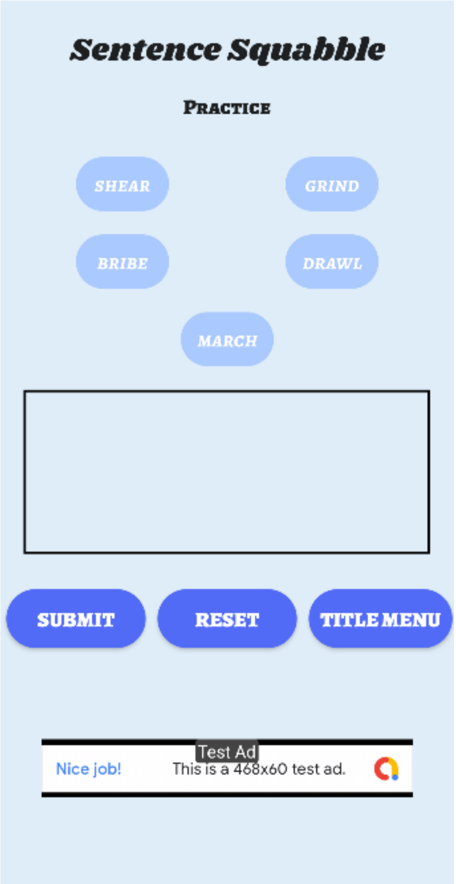


Figure 10: Prototype Design - Practice Page

5 Verification and Validation

5.1 Prototype Performance Metrics

The game's performance was benchmarked in four general categories: Computational Performance, Fun, Social Engagement, and Monetary. The general strategy for evaluating the performance of the game was facilitated by surveys.

The evaluation of the computational performance of a game like Sentence Squabble was determined to be simple since the expectation is that there is no perceivable delay by the user, in other words the experience would need to be seamless. To evaluate the player's perception of seamlessness, play testers were given a survey question inquiring about the responsiveness of the application.

To evaluate how fun the game is, fun was broken down into three sub-categories, visual pleasure, amusement from the generated words, and amusement from building the sentence. Since fun is a feeling and subjective the play testers were prompted to rate the applications visual pleasure, amusement from the generated words, and the amusement from building the sentence on a scale from 1 to 4.

Social engagement was difficult to measure in a prototype setting, however, to get an estimate a survey question asking the play testers how likely they are likely to share their created sentence with others on a scale from 1 to 4 was created.

The monetary pillar was also determined utilizing a survey to gauge play tester's feeling on the advertisements presented to them. The goal was to make the advertisements noticeable enough to catch the player's eye but not so much that they appear intrusive to the game experience. We asked the play testers how impactful the advertisement was on their gameplay experience on a scale from 1 to 4.

Finally, in our survey we left a general prompt for feedback on the game in general to allow play testers to provide additional comments on their experience.

The survey, https://forms.gle/PN1xxB5eDHj8hVh66, was created to give to play testers after they had tested the application.

5.2 Requirements Fulfillment Analysis

5.3 Application Performance

The application was deployed on a Samsung S22, opposed to the emulated Pixel 2 used during development, for a realistic performance comparison to similar word applications. Due to legal, practical, and access limitations for benchmarking similar applications consisting of proprietary software a subjective evaluation of performance was conducted. The Sentence Squabble application was found to have similar command, navigation, response times compared to similar word game applications such as Quordle and Wordle. However, two points of deviation were identified, one was the time required for Sentence Squabble to start, which was far greater than similar applications, and the second was advertisement loading times.

The difference in startup time was attributed to word generation being done on the device instead of by a server and synchronized to the application. Therefore, a deployable version of Sentence Squabble would require a server to provide the word generation and synchronization of the daily challenge words, and any required practice words, between the application and the server.

The advertisement loading times resulted from a failure to buffer the practice and challenge mode views at startup. Instead of having the game modes pre-loaded visually they were recreated every time the user navigated to a particular game mode. Since the game modes are light weight in themselves there was no appreciable load time for the rest of the game mode view however the advertisement would pop in a few moments after the view loaded. This would be remedied for a deployable version of Sentence Squabble by simply ensuring that the game modes were initialized and buffered when the application is opened instead of being recreated every time the user switches game modes.

5.4 Game Testing Survey Results

The game was tested by ten individuals who were given a survey, https://forms.gle/PN1xxB5eDHj8hVh66, to fill out after their play session. The survey results were analyzed and summarized in the figures below. Overall, the survey revealed that the core gameplay concept of using randomly generated words to create a sentence is viable and that the visual aesthetic of the game was enjoyed by our play testers. However, improvements in the words presented and player feedback on failed attempts need improvement. Additionally, it seems that players were satisfied with how advertisements were implemented and did not find that they were intrusive to their overall experience and were likely to recommend the application to their friends or family. Play testers were also provided with an empty prompt to provide additional feedback if they desired. The feedback revolved around improving the grammar checking and player feedback on failed attempts as well as the expression of the desire for multiple difficulty levels of play.

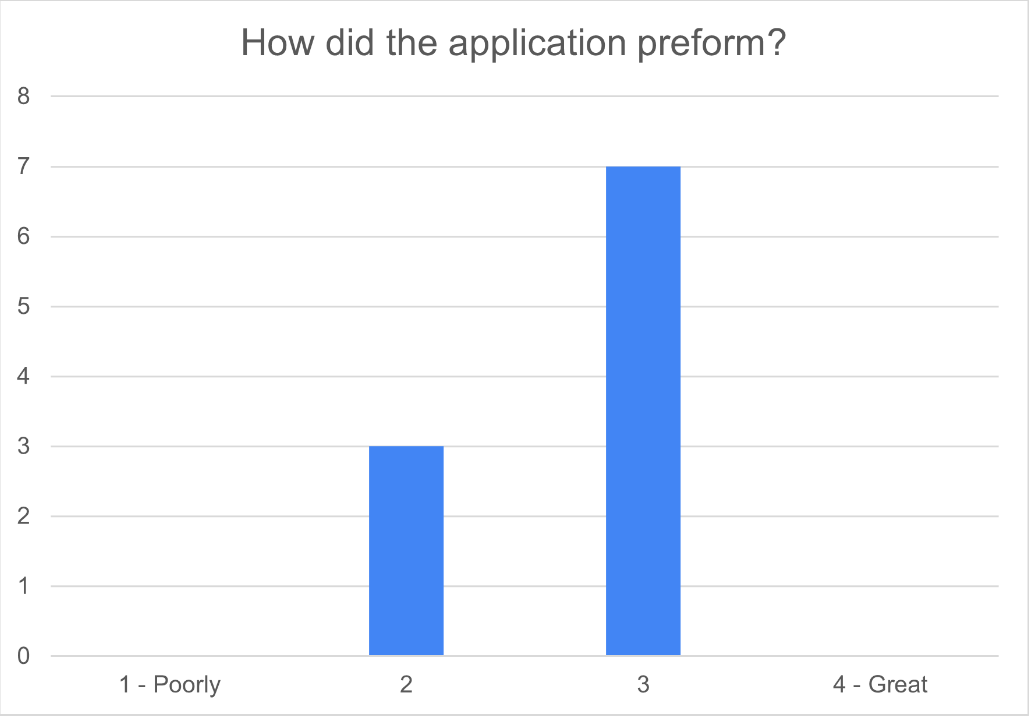


Figure 11: Survey - Application Performance

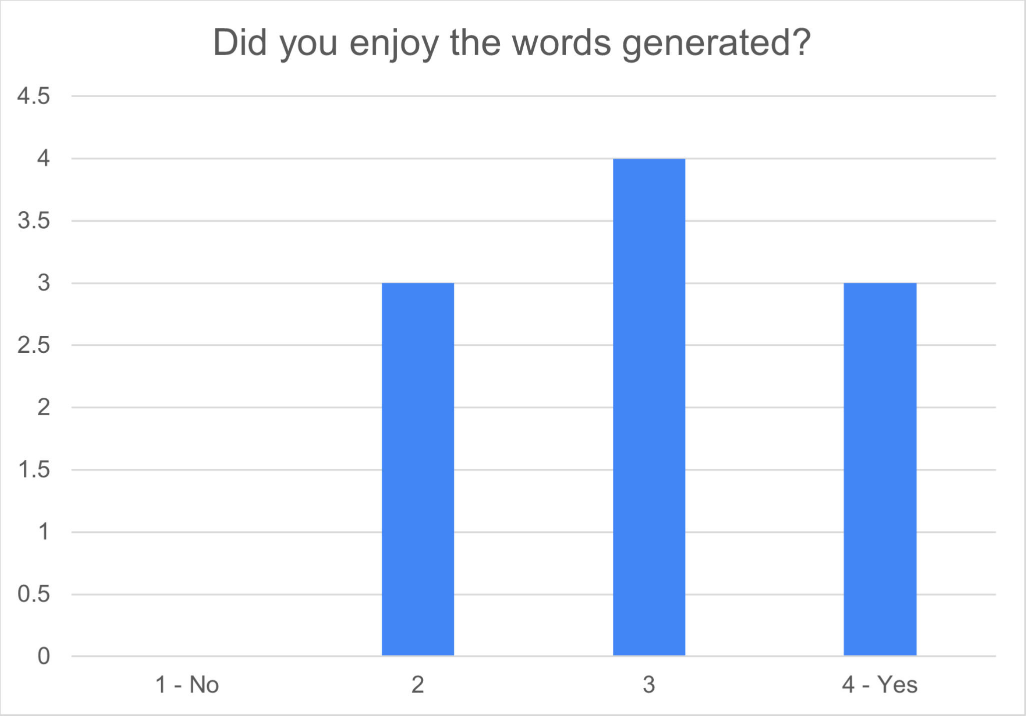


Figure 12: Survey - Word Enjoyment

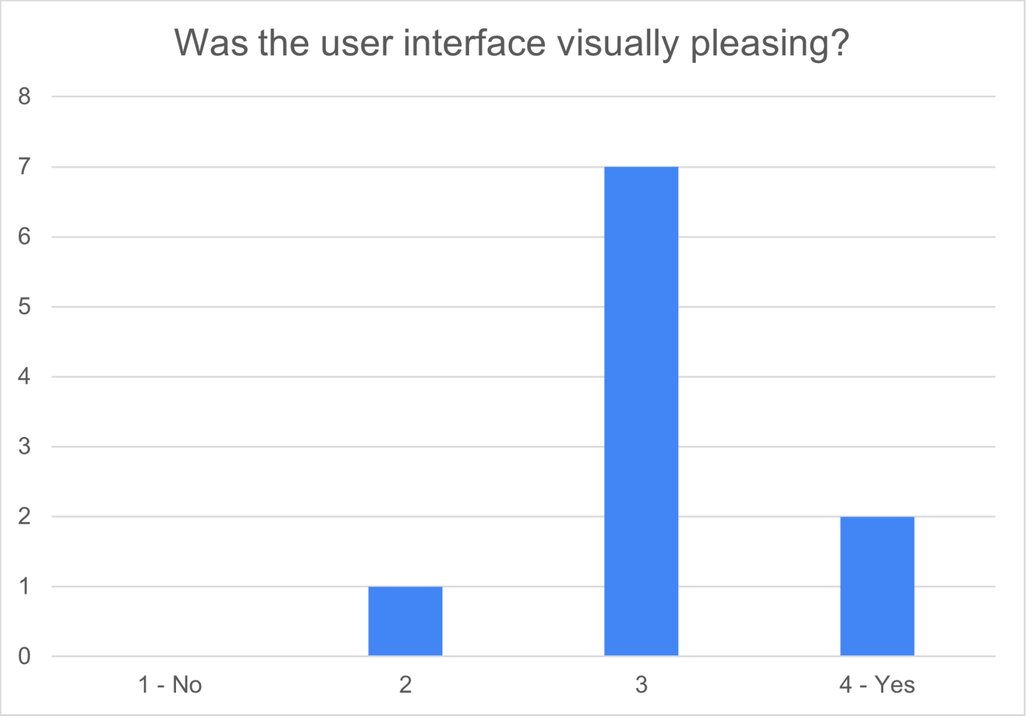


Figure 13: Survey - Visually Pleasing

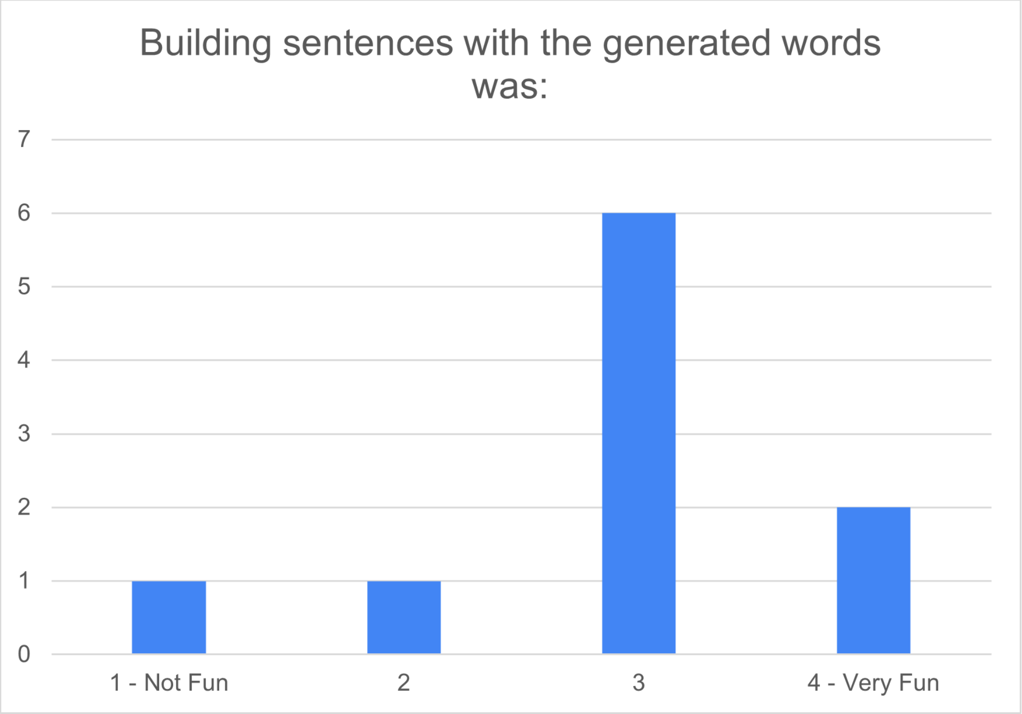


Figure 14: Survey - Sentence Building Fun

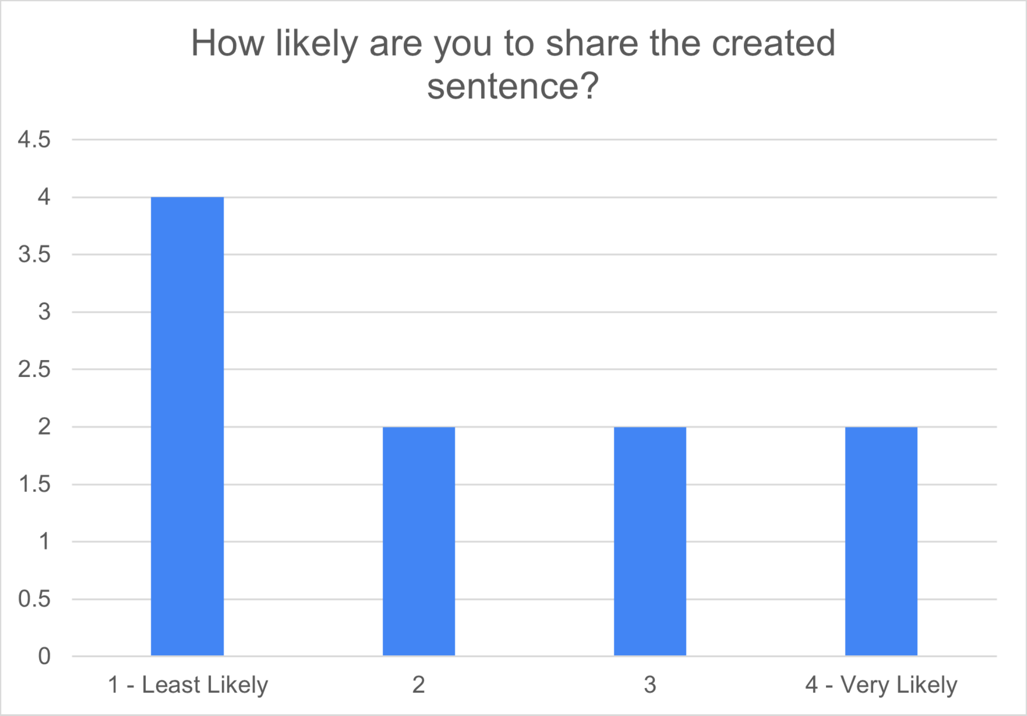


Figure 15: Survey - Sharing Likelyhood

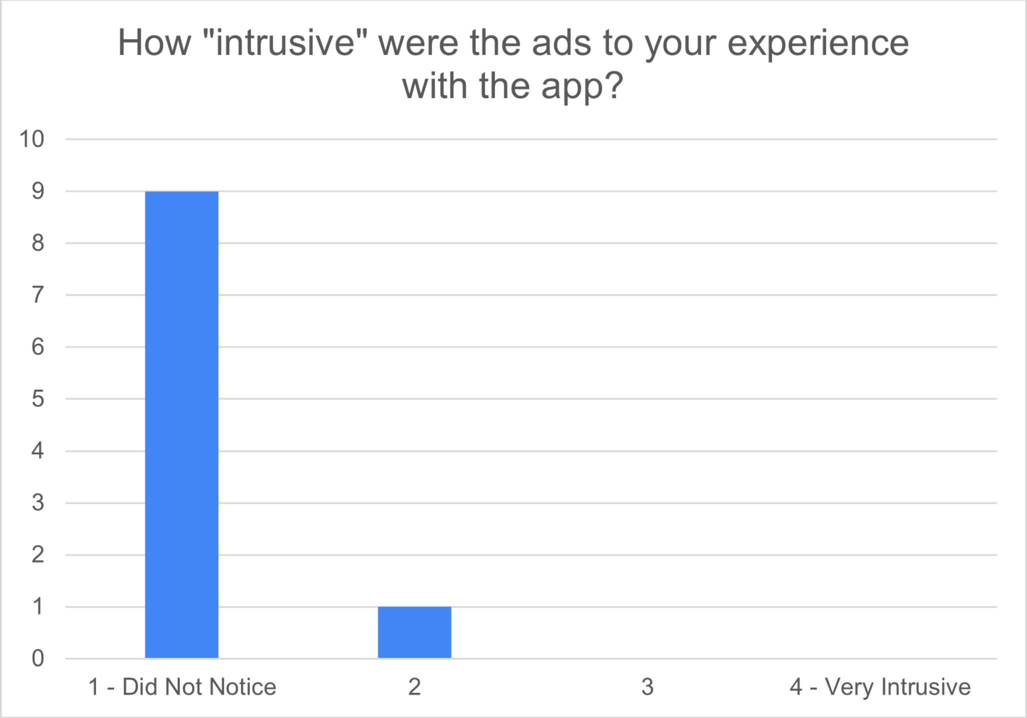


Figure 16: Survey - Advertisement Intrusiveness

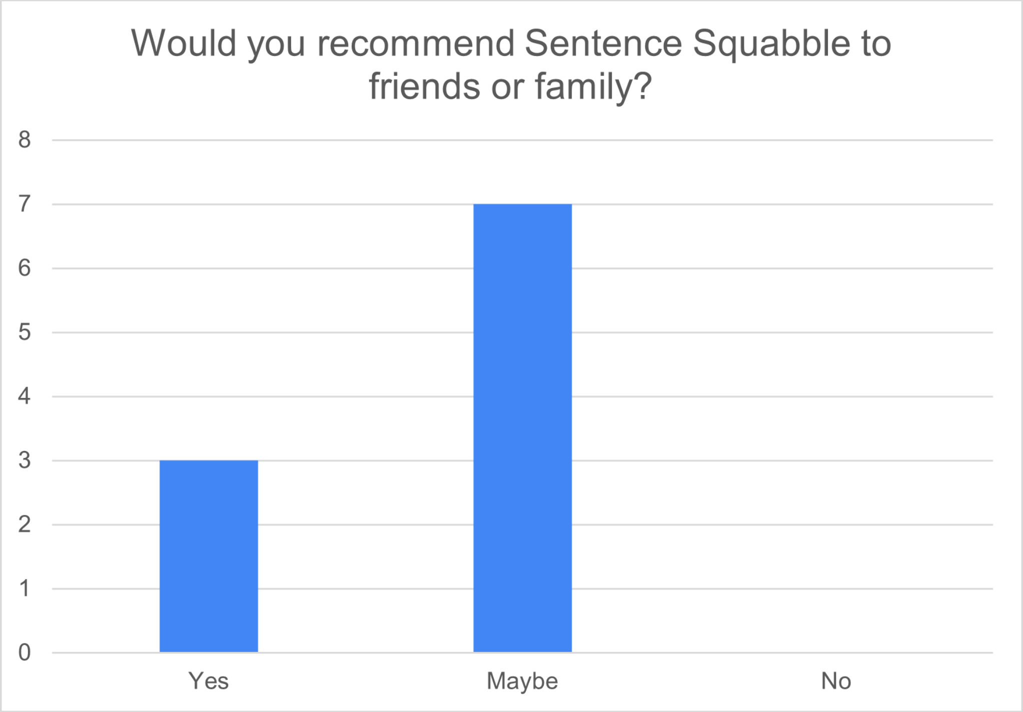


Figure 17: Survey - Recommendation to Others

5.5 Required Improvements for Production Product

Based upon the feedback from play testers, and our own knowledge of the application, the following list was compiled for required improvements to allow Sentence Squabble to be a viable commercial product:

1. Improve player feedback.
    - Develop a method for providing players with the definitions of the words they must use.
    - Provide some amount of feedback on why the player's sentence was not accepted.
    - More pronounced failed state when player's sentence was not accepted.
2. UI
    - A font that more clearly indicates upper vs. lower case words so words that must be capitalized to be valid are clearly displayed.
3. Backend
    - Pre-load game modes to ensure advertisements do not "pop-in" when player changes game modes.
    - Implement grammar checking server with third party service since on-device grammar checking is not feasible.
    - Implement dictionary server with third party service to prevent long loading times while dictionary is downloaded.
4. Game Design
    - Improve word vetting/filtering to provide a better player experience.

- Improve sentence grammar checking capabilities.

6 Conclusion

Through this quarter the game Sentence Squabble was planned, analyzed, prototyped, and tested to determine if it meets the predicted metrics, would produce the expected benefits, and would constitute a viable product. The prototype was successful in providing a vertical slice of the intended game play and was positively received by our test audience resulting in this team deducing that Sentence Squabble achieved all preconceived objectives. Although additional development would be necessary to bring this product to market a clear set of required improvements has been identified and additional costs identified.

Appendices

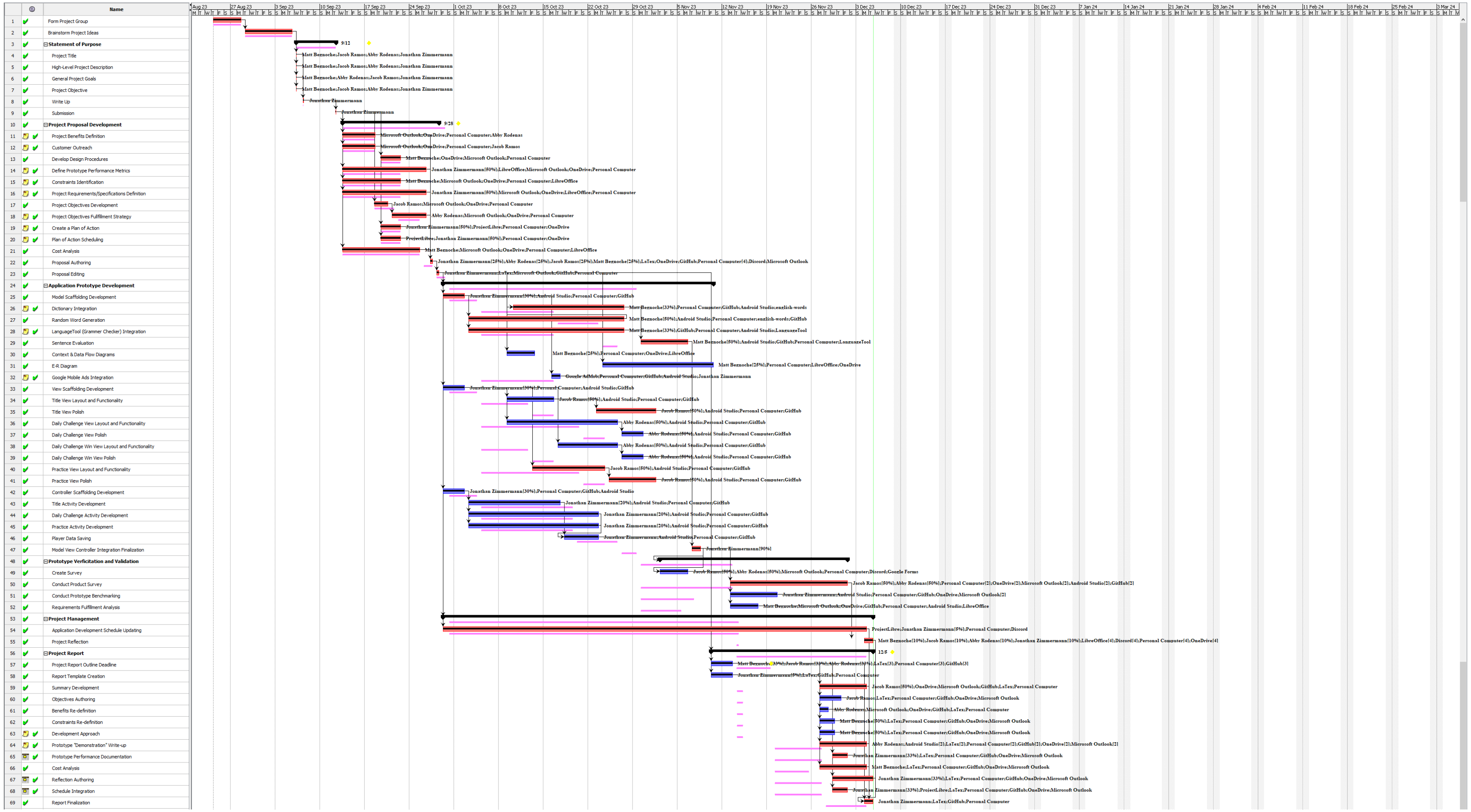
A Detailed Project Schedule

Table 5: Project Task List

ID	Name	Resource
1	Form Project Group	Matt Begnoche;Jacob Ramos;Abby Rodenas;Jonathan Zimmermann
2	Brainstorm Project Ideas	Matt Begnoche;Jacob Ramos;Abby Rodenas;Jonathan Zimmermann
3	Statement of Purpose	N/A - Milestone
4	Project Title	Matt Begnoche;Jacob Ramos;Abby Rodenas;Jonathan Zimmermann
5	High-Level Project Description	Matt Begnoche;Jacob Ramos;Abby Rodenas;Jonathan Zimmermann
6	General Project Goals	Matt Begnoche;Abby Rodenas;Jacob Ramos;Jonathan Zimmermann
7	Project Objective	Matt Begnoche;Jacob Ramos;Abby Rodenas;Jonathan Zimmermann
8	Write Up	Jonathan Zimmermann
9	Submission	Jonathan Zimmermann
10	Project Proposal Development	N/A - Milestone
11	Project Benefits Definition	Microsoft Outlook;OneDrive;Personal Computer;Abby Rodenas
12	Customer Outreach	Microsoft Outlook;OneDrive;Personal Computer;Jacob Ramos
13	Develop Design Procedures	Matt Begnoche;OneDrive;Microsoft Outlook;Personal Computer
14	Define Prototype Performance	Jonathan Zimmermann[50%];LibreOffice;Microsoft Outlook;OneDrive;Personal Computer
15	Constraints Identification	Matt Begnoche;Microsoft Outlook;OneDrive;Personal Computer;LibreOffice
16	Project Requirements/Specifications Definition	Jonathan Zimmermann[50%];Microsoft Outlook;OneDrive;LibreOffice;Personal Computer
17	Project Objectives Development	Jacob Ramos;Microsoft Outlook;OneDrive;Personal Computer
18	Project Objectives Fullfillment Strategy	Abby Rodenas;Microsoft Outlook;OneDrive;Personal Computer
19	Create a Plan of Action	Jonathan Zimmermann[50%];ProjectLibre;Personal Computer;OneDrive
20	Plan of Action Scheduling	ProjectLibre;Jonathan Zimmermann[50%];Personal Computer;OneDrive
21	Cost Analysis	Matt Begnoche;Microsoft Outlook;OneDrive;Personal Computer;LibreOffice
22	Proposal Authoring	Jonathan Zimmermann[25%];Abby Rodenas[25%];Jacob Ramos[25%];Matt Begnoche[25%];LaTex;OneDrive;GitHub;Personal Computer[4];Discord;Microsoft Outlook
23	Proposal Editing	Jonathan Zimmermann;LaTex;Microsoft Outlook;GitHub;Personal Computer
24	Application Prototype Development	N/A - Milestone
25	Model Scaffolding Development	Jonathan Zimmermann[30%];Android Studio;Personal Computer;GitHub
26	Dictionary Integration	Matt Begnoche[33%];Personal Computer;GitHub;Android Studio;english-words
27	Random Word Generation	Matt Begnoche[50%];Android Studio;Personal Computer;english-words;GitHub
28	LanguageTool (Grammer Checker) Integration	Matt Begnoche[33%];GitHub;Personal Computer;Android Studio;LanguageTool
29	Sentence Evaluation	Matt Begnoche[50%];Android Studio;GitHub;Personal Computer;LanguageTool
30	Context & Data Flow Diagrams	Matt Begnoche[25%];Personal Computer;OneDrive;LibreOffice
31	E-R Diagram	Matt Begnoche[25%];Personal Computer;LibreOffice;OneDrive
32	Google Mobile Ads Integration	Google AdMob;Personal Computer;GitHub;Android Studio;Jonathan Zimmermann
33	View Scaffolding Development	Jonathan Zimmermann[30%];Personal Computer;Android Studio;GitHub
34	Title View Layout and Functionality	Jacob Ramos[50%];Android Studio;Personal Computer;GitHub
35	Title View Polish	Jacob Ramos[50%];Android Studio;Personal Computer;GitHub
36	Daily Challenge View Layout and Functionality	Abby Rodenas[50%];Android Studio;Personal Computer;GitHub
37	Daily Challenge View Polish	Abby Rodenas[50%];Android Studio;Personal Computer;GitHub
38	Daily Challenge Win View Layout and Functionality	Abby Rodenas[50%];Android Studio;Personal Computer;GitHub
39	Daily Challenge Win View Polish	Abby Rodenas[50%];Android Studio;Personal Computer;GitHub
40	Practice View Layout and Functionality	Jacob Ramos[50%];Android Studio;Personal Computer;GitHub
41	Practice View Polish	Jacob Ramos[50%];Android Studio;Personal Computer;GitHub
42	Controller Scaffolding Development	Jonathan Zimmermann[30%];Personal Computer;GitHub;Android Studio
43	Title Activity Development	Jonathan Zimmermann[20%];Android Studio;Personal Computer;GitHub
44	Daily Challenge Activity Development	Jonathan Zimmermann[20%];Android Studio;Personal Computer;GitHub
45	Practice Activity Development	Jonathan Zimmermann[20%];Android Studio;Personal Computer;GitHub
46	Player Data Saving	Jonathan Zimmermann;Android Studio;Personal Computer;GitHub
47	Model View Controller Integration Finalization	Jonathan Zimmermann[90%]
48	Prototype Verficitation and Validation	N/A - Milestone
49	Create Survey	Jacob Ramos[50%];Abby Rodenas[50%];Microsoft Outlook;Personal Computer;Discord;Google Forms
50	Conduct Product Survey	Jacob Ramos[50%];Abby Rodenas[50%];Personal Computer[2];OneDrive[2];Microsoft Outlook[2];Android Studio[2];GitHub[2]
51	Conduct Prototype Benchmarking	Jonathan Zimmermann;Android Studio;Personal Computer;GitHub;OneDrive;Microsoft Outlook[2]
52	Requirements Fulfillment Analysis	Matt Begnoche;Microsoft Outlook;OneDrive;GitHub;Personal Computer;Android Studio;LibreOffice
53	Project Management	N/A - Milestone
54	Application Development Schedule Updating	ProjectLibre;Jonathan Zimmermann[5%];Personal Computer;Discord
55	Project Reflection	Matt Begnoche[10%];Jacob Ramos[10%];Abby Rodenas[10%];Jonathan Zimmermann[10%];LibreOffice[4];Discord[4];Personal Computer[4];OneDrive[4]
56	Project Report	N/A - Milestone
57	Project Report Outline Deadline	Matt Begnoche[33%];Jacob Ramos[33%];Abby Rodenas[33%];LaTex[3];Personal Computer[3];GitHub[3]
58	Report Template Creation	Jonathan Zimmermann[5%];LaTex;GitHub;Personal Computer
59	Summary Development	Jacob Ramos[50%];OneDrive;Microsoft Outlook;GitHub;LaTex;Personal Computer
60	Objectives Authoring	Jacob Ramos;LaTex;Personal Computer;GitHub;OneDrive;Microsoft Outlook
61	Benefits Re-definition	Abby Rodenas;Microsoft Outlook;OneDrive;GitHub;LaTex;Personal Computer
62	Constraints Re-definition	Matt Begnoche[50%];LaTex;Personal Computer;GitHub;OneDrive;Microsoft Outlook
63	Development Approach	Matt Begnoche[50%];LaTex;Personal Computer;GitHub;OneDrive;Microsoft Outlook
64	Prototype Demonstration Write-Up	Abby Rodenas;Android Studio[2];LaTex[2];Personal Computer[2];GitHub[2];OneDrive[2];Microsoft Outlook[2]
65	Prototype Performance Documentation	Jonathan Zimmermann[33%];LaTex;Personal Computer;GitHub;OneDrive;Microsoft Outlook
66	Cost Analysis	Matt Begnoche;LaTex;Personal Computer;GitHub;OneDrive;Microsoft Outlook
67	Reflection Authoring	Jonathan Zimmermann[33%];LaTex;Personal Computer;GitHub;OneDrive;Microsoft Outlook
68	Schedule Integration	Jonathan Zimmermann[33%];ProjectLibre;LaTex;Personal Computer;GitHub;OneDrive;Microsoft Outlook
69	Report Finalization	Jonathan Zimmermann;LaTex;GitHub;Personal Computer

Table 6: Project Task Tracking

ID	Name	Actual Start	Actual Finish	Actual Work
1	Form Project Group	8/24/2023 8:00	8/28/2023 17:00	40 hours
2	Brainstorm Project Ideas	8/29/2023 8:00	9/5/2023 17:00	64 hours
3	Statement of Purpose	9/6/2023 8:00	9/12/2023 13:48	34.8 hours
4	Project Title	9/6/2023 8:00	9/6/2023 10:00	8 hours
5	High-Level Project Description	9/6/2023 8:00	9/6/2023 10:00	8 hours
6	General Project Goals	9/6/2023 8:00	9/6/2023 10:00	8 hours
7	Project Objective	9/6/2023 8:00	9/6/2023 10:00	8 hours
8	Write Up	9/7/2023 10:00	9/7/2023 13:00	2 hours
9	Submission	9/12/2023 12:00	9/12/2023 13:48	0.8 hours
10	Project Proposal Development	9/13/2023 13:48	9/28/2023 17:00	570.8 hours
11	Project Benefits Definition	9/13/2023 13:48	9/18/2023 17:00	43.2 hours
12	Customer Outreach	9/13/2023 13:48	9/18/2023 17:00	43.2 hours
13	Develop Design Procedures	9/19/2023 13:48	9/22/2023 17:00	25.6 hours
14	Define Prototype Performance	9/13/2023 13:48	9/26/2023 17:00	83.68 hours
15	Constraints Identification	9/13/2023 13:48	9/22/2023 17:00	75.2 hours
16	Project Requirements/Specifications Definition	9/13/2023 13:48	9/26/2023 17:00	79.92 hours
17	Project Objectives Development	9/18/2023 13:48	9/20/2023 17:00	19.2 hours
18	Project Objectives Fullfillment Strategy	9/21/2023 8:00	9/26/2023 17:00	48 hours
19	Create a Plan of Action	9/19/2023 13:48	9/22/2023 17:00	0 hours
20	Plan of Action Scheduling	9/19/2023 13:48	9/22/2023 17:00	13.6 hours
21	Cost Analysis	9/13/2023 13:48	9/25/2023 17:00	99.2 hours
22	Proposal Authoring	9/27/2023 8:00	9/27/2023 17:00	32 hours
23	Proposal Editing	9/28/2023 8:00	9/28/2023 17:00	8 hours
24	Application Prototype Development	9/29/2023 8:00	11/10/2023 17:00	1768 hours
25	Model Scaffolding Development	9/29/2023 8:00	10/2/2023 17:00	32 hours
26	Dictionary Integration	10/10/2023 8:00	10/27/2023 17:00	144 hours
27	Random Word Generation	10/3/2023 8:00	10/27/2023 17:00	200 hours
28	LanguageTool (Grammer Checker) Integration	10/3/2023 8:00	10/27/2023 17:00	200 hours
29	Sentence Evaluation	10/30/2023 8:00	11/6/2023 17:00	32 hours
30	Context & Data Flow Diagrams	10/9/2023 8:00	10/15/2023 17:00	56 hours
31	E-R Diagram	10/24/2023 8:00	11/10/2023 17:00	144 hours
32	Google Mobile Ads Integration	10/16/2023 8:00	10/17/2023 17:00	16 hours
33	View Scaffolding Development	9/29/2023 8:00	10/2/2023 17:00	32 hours
34	Title View Layout and Functionality	10/9/2023 8:00	10/16/2023 17:00	24 hours
35	Title View Polish	10/23/2023 8:00	11/1/2023 17:00	80 hours
36	Daily Challenge View Layout and Functionality	10/9/2023 8:00	10/26/2023 17:00	144 hours
37	Daily Challenge View Polish	10/27/2023 8:00	10/30/2023 17:00	8 hours
38	Daily Challenge Win View Layout and Functionality	10/17/2023 8:00	10/26/2023 17:00	80 hours
39	Daily Challenge Win View Polish	10/27/2023 8:00	10/30/2023 17:00	8 hours
40	Practice View Layout and Functionality	10/13/2023 8:00	10/24/2023 17:00	96 hours
41	Practice View Polish	10/25/2023 8:00	11/1/2023 17:00	64 hours
42	Controller Scaffolding Development	9/29/2023 8:00	10/2/2023 17:00	32 hours
43	Title Activity Development	10/3/2023 8:00	10/17/2023 17:00	24 hours
44	Daily Challenge Activity Development	10/3/2023 8:00	10/23/2023 17:00	152 hours
45	Practice Activity Development	10/3/2023 8:00	10/23/2023 17:00	152 hours
46	Player Data Saving	10/18/2023 8:00	10/23/2023 17:00	32 hours
47	Model View Controller Integration Finalization	11/7/2023 8:00	11/8/2023 17:00	16 hours
48	Prototype Verficitation and Validation	11/2/2023 8:00	12/1/2023 17:00	320 hours
49	Create Survey	11/2/2023 8:00	11/6/2023 17:00	64 hours
50	Conduct Product Survey	11/13/2023 8:00	12/1/2023 17:00	152 hours
51	Conduct Prototype Benchmarking	11/13/2023 8:00	11/20/2023 17:00	64 hours
52	Requirements Fulfillment Analysis	11/13/2023 8:00	11/17/2023 17:00	40 hours
53	Project Management	9/29/2023 8:00	12/5/2023 17:00	516.8 hours
54	Application Development Schedule Updating	9/29/2023 8:00	12/4/2023 17:00	504 hours
55	Project Reflection	12/2/2023 8:00	12/5/2023 17:00	12.8 hours
56	Project Report	11/10/2023 8:00	12/5/2023 17:00	347.6 hours
57	Project Report Outline Deadline	11/10/2023 8:00	11/13/2023 17:00	31.68 hours
58	Report Template Creation	11/10/2023 8:00	11/13/2023 17:00	1.6 hours
59	Summary Development	11/27/2023 8:00	12/4/2023 17:00	64 hours
60	Objectives Authoring	11/27/2023 8:00	11/30/2023 17:00	32 hours
61	Benefits Re-definition	11/27/2023 8:00	11/28/2023 17:00	16 hours
62	Constraints Re-definition	11/27/2023 8:00	11/29/2023 17:00	12 hours
63	Development Approach	11/27/2023 8:00	11/29/2023 17:00	12 hours
64	Prototype Demonstration Write-Up	11/27/2023 8:00	12/4/2023 17:00	64 hours
65	Prototype Performance Documentation	11/29/2023 8:00	12/1/2023 17:00	7.92 hours
66	Cost Analysis	11/27/2023 8:00	12/4/2023 17:00	64 hours
67	Reflection Authoring	11/29/2023 8:00	12/5/2023 17:00	18.48 hours
68	Schedule Integration	11/29/2023 8:00	12/1/2023 17:00	7.92 hours
69	Report Finalization	12/4/2023 8:00	12/5/2023 17:00	16 hours



		Name
1	✓	Form Project Group
2	✓	Brainstorm Project Ideas
3	✓	⊟ **Statement of Purpose**
4	✓	Project Title
5	✓	High-Level Project Description
6	✓	General Project Goals
7	✓	Project Objective
8	✓	Write Up
9	✓	Submission
10	✓	⊟ **Project Proposal Development**
11	🗒 ✓	Project Benefits Definition
12	🗒 ✓	Customer Outreach
13	✓	Develop Design Procedures
14	🗒 ✓	Define Prototype Performance Metrics
15	🗒 ✓	Constraints Identification
16	🗒 ✓	Project Requirements/Specifications Definition
17	✓	Project Objectives Development
18	🗒 ✓	Project Objectives Fullfillment Strategy
19	🗒 ✓	Create a Plan of Action
20	🗒 ✓	Plan of Action Scheduling
21	✓	Cost Analysis
22	✓	Proposal Authoring
23	✓	Proposal Editing
24	✓	⊟ **Application Prototype Development**
25	✓	Model Scaffolding Development
26	🗒 ✓	Dictionary Integration
27	✓	Random Word Generation
28	🗒 ✓	LanguageTool (Grammer Checker) Integration
29	✓	Sentence Evaluation
30	✓	Context & Data Flow Diagrams
31	✓	E-R Diagram
32	🗒 ✓	Google Mobile Ads Integration
33	✓	View Scaffolding Development
34	✓	Title View Layout and Functionality
35	✓	Title View Polish
36	✓	Daily Challenge View Layout and Functionality
37	✓	Daily Challenge View Polish
38	✓	Daily Challenge Win View Layout and Functionality
39	✓	Daily Challenge Win View Polish
40	✓	Practice View Layout and Functionality
41	✓	Practice View Polish
42	✓	Controller Scaffolding Development
43	✓	Title Activity Development
44	✓	Daily Challenge Activity Development
45	✓	Practice Activity Development
46	✓	Player Data Saving
47	✓	Model View Controller Integration Finalization
48	✓	⊟ **Prototype Verficitation and Validation**
49	✓	Create Survey
50	✓	Conduct Product Survey
51	✓	Conduct Prototype Benchmarking
52	✓	Requirements Fulfillment Analysis
53	✓	⊟ **Project Management**
54	✓	Application Development Schedule Updating
55	✓	Project Reflection
56	✓	⊟ **Project Report**
57	✓	Project Report Outline Deadline
58	✓	Report Template Creation
59	✓	Summary Development
60	✓	Objectives Authoring
61	✓	Benefits Re-definition
62	✓	Constraints Re-definition
63	🗒 ✓	Development Approach
64	🗒 ✓	Prototype "Demonstration" Write-up
65	📅 ✓	Prototype Performance Documentation
66	✓	Cost Analysis
67	📅 ✓	Reflection Authoring
68	📅 ✓	Schedule Integration
69	✓	Report Finalization

Figure 18: Project Gantt Chart