

EASY GRAMMAR

20/02/24

PROJECT MILESTONE 1 REPORT

TEAM MEMBERS:

YUJIE PENG,
XAVIER PADANAUD,
TingHin Cheung,
RAVINESH SAMI

Git repository: <https://github.com/RileyC9/EasyGrammar>

Grammar should be easy and fun.

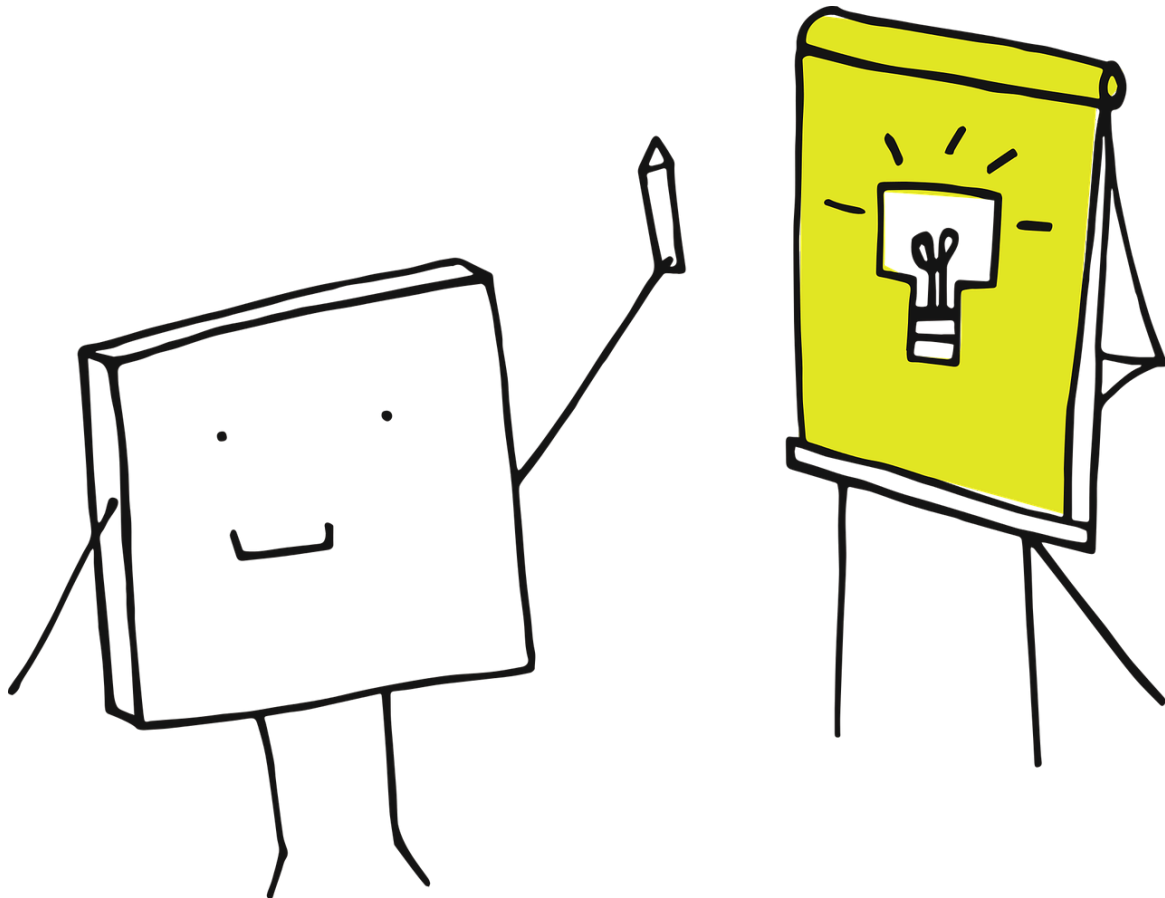
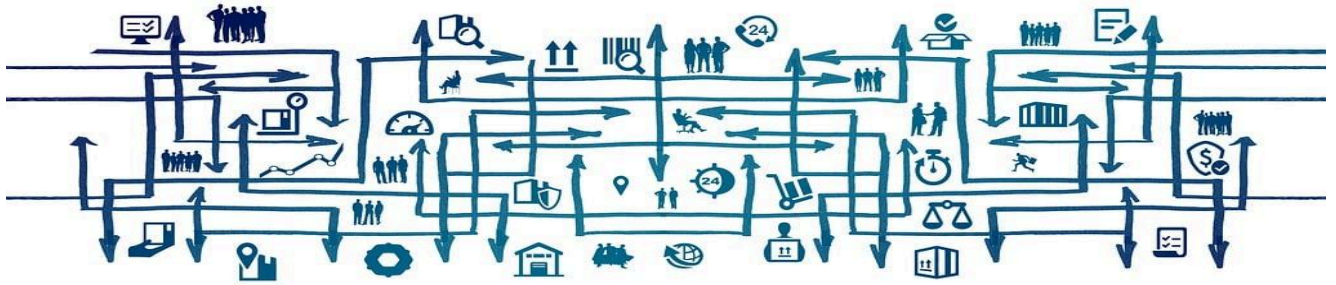


Table of Contents

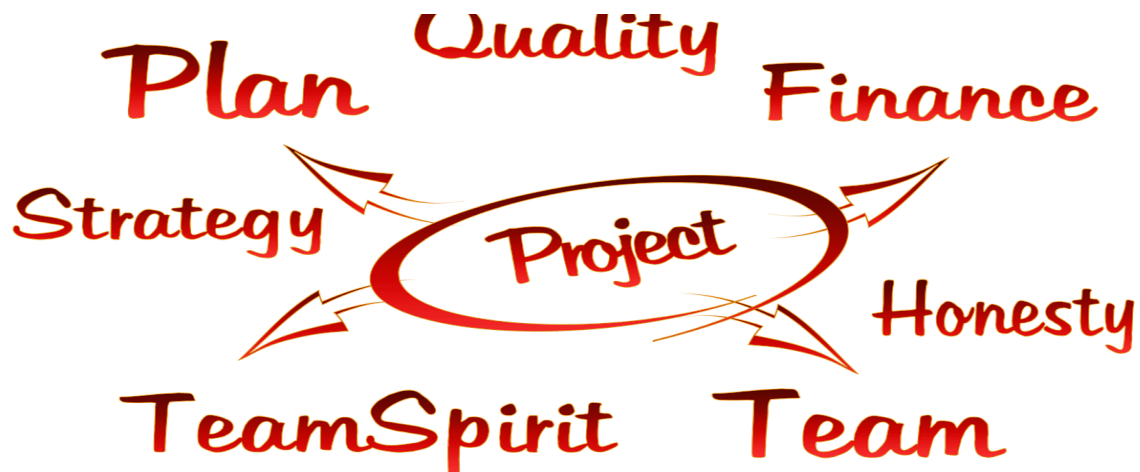
| | |
|------------------------------------|----|
| Project overview..... | 3 |
| SDLC..... | 4 |
| User Stories..... | 5 |
| Technology Stack..... | 7 |
| APIs and Features..... | 8 |
| WBS and Project Schedule..... | 9 |
| Project Timeline Preview..... | 9 |
| WBS Preview..... | 10 |
| Wireframes..... | 11 |
| Prototype..... | 11 |
| Data Flow Diagram..... | 12 |
| Reference..... | 15 |
| Related Links..... | 16 |
| Appendix:..... | 17 |
| Communication..... | 17 |
| More on Git branch management..... | 17 |
| Swimlane diagram..... | 18 |
| WBS..... | 19 |

Project overview



Our project aims to develop an application that will assist users in improving their English language skills. These skills include writing style, grammar, pronunciation, and sentence structure. This application will use the combination of dictionary functionality , AI-generated image, and AI evaluation on the user description. The application will integrate two main components: Free Dictionary API and an OpenAI API.

SDLC



We have chosen Feature Driven Development, which is a practical Agile approach. As the name suggests, it focuses on developing working software with features that satisfy client needs. FDD also focuses on delivering features incrementally. This model aligns well with our project goals, allowing us to prioritize and develop essential features iteratively, ensuring steady progress while we complete features one by one and timely delivery. Additionally, FDD emphasizes collaboration and communication among teams, fostering a cohesive development process.

User Stories

| API | User Story | Persona | |
|-------------------------------------|--|--|--|
| Dictionary API | As an International Student in English literature, I want to improve my English vocabulary, learn definitions so that I can get good grades on my courses. | Meet Ramcharan, the international student! | |
| | | Demographics | Age: 20 Education: Currently in University, enrolled under English Literature |
| | | Background | <ul style="list-style-type: none"> From India Works part-time at a retail shop |
| | | Characteristics | <ul style="list-style-type: none"> Not fluent in English Relies on apps to learn the English language. Finds it hard to communicate with his customers at work and his friends at school |
| | | Goals | <ul style="list-style-type: none"> Want to improve his English vocabulary and sentence structure. Communicate effectively with his customers and friends. Get good grades in his English courses. |
| | As an entrepreneur travelling to English-speaking countries, I want to enhance my vocabulary with better pronunciation, so that I can communicate with my international clients and partners more effectively. | Meet Chang, the Entrepreneur! | |
| | | Demographics | <ul style="list-style-type: none"> Age: 30 Occupation: Entrepreneur |
| | | Background | <ul style="list-style-type: none"> From China Travels to English speaking countries frequently |
| | | Characteristics | <ul style="list-style-type: none"> Not able to pronounce English words correctly. Relies on apps to learn the English language. Finds it hard to communicate with his international clients and partners. |
| | | Goals | <ul style="list-style-type: none"> Want to improve his English pronunciation. Communicate effectively with his clients. Increase his business in other English-speaking countries. |

| | | | |
|-------------------|---|---|--|
| OpenAI API | As an International Student in Computer Science, I want to improve my English grammar and writing skills so that I can effectively communicate my ideas and concepts. | Meet Javier, the international student! | |
| | | Demographics | <ul style="list-style-type: none"> Age: 25 Education: Currently in University, enrolled under Computer Science |
| | | Background | <ul style="list-style-type: none"> From Mexico Works part-time at an IT company |
| | | Characteristics | <ul style="list-style-type: none"> Has problems writing reports, emails for his work and school. Has grammatical errors in his sentences. Relies on apps to improve his English writing skills. |
| | | Goals | <ul style="list-style-type: none"> Want to improve his English writing skills and sentence structure. Show professionalism in English at his work so that he has a good future in the company. |
| | As a travel blogger, I want to enhance my English writing skills, so that I can attract more readers to my blog. | Meet Mei, a travel blogger! | |
| | | Demographics | <ul style="list-style-type: none"> Age: 29 Occupation: Explorer |
| | | Background | <ul style="list-style-type: none"> From Japan Travels around the world to explore different cultures and document her experiences. |
| | | Characteristics | <ul style="list-style-type: none"> Has problems with writing English sentences. Relies on apps to learn English writing skills. Feels reluctant to share her post due to grammatical errors. |
| | | Goals | <ul style="list-style-type: none"> Want to improve her English writing skills. Improve on grammatical errors. Get as many readers as possible to read her blogs. |

Technology Stack



We have chosen **React.js** for frontend, and **Netlify** for deployment.

React offers a flexible and efficient framework for building interactive user interfaces, and manipulating API responses displaying on the browser. Netlify simplifies the deployment process by offering easy integration with our Git repository and providing continuous deployment capabilities. This technology stack aligns well with our project's requirements, offering a combination of performance, reliability, and ease of use.

APIs and Features

1. **Free Dictionary API:** We chose dictionary API for its comprehensive word database and reliable word validation features. Also, for providing sample sentences.

Features:

1. Provide definitions: Users can search a vocabulary and get the definition.
 2. Sample sentence retrieval: Users can learn sample sentences.
 3. Pronunciation: Users can listen to the pronunciation of the word.
2. **OpenAI API:** We selected OpenAI API for its advanced image generation and text analyzing capabilities.

Features:

1. Image generation and display: Users can see an image generated from the sample sentences.
2. Check for grammar and sentence correctness: Users' description will be sent to OpenAI API for grammar analysis.
3. Feedback and scoring mechanisms: Users will receive customised feedback based on the grammar mistakes they made.

[illegible]

WBS Preview


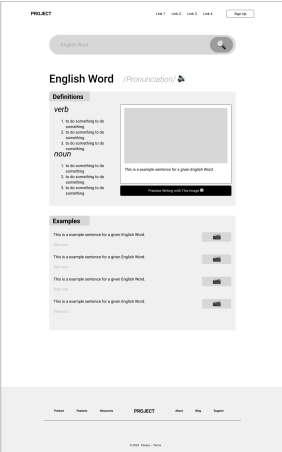
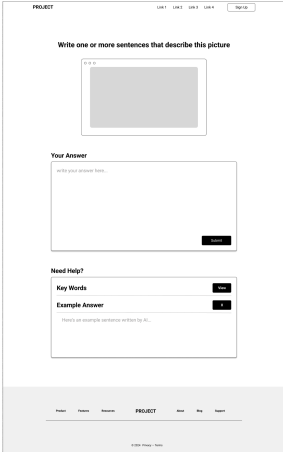
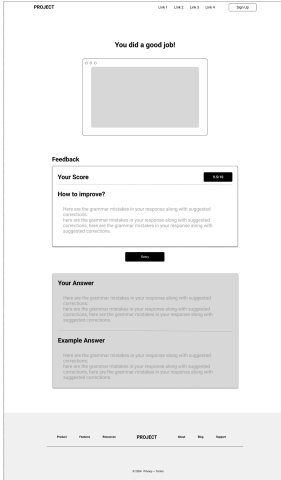
Project Schedule : Easy Grammar

WEEK OF

21/02/2024

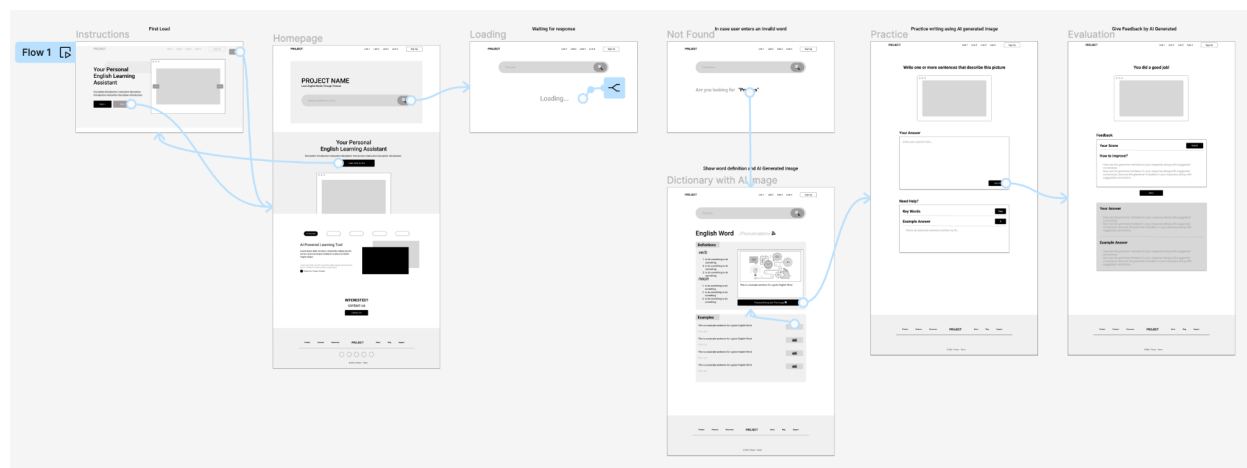
| Task # | Task | Assigned To | Estimated Hours | Actual Hours | Comments/ Estimated complete |
|------------|---|-------------|-----------------|--------------|---------------------------------|
| 1 | General Requirement | | | | |
| 1.1 | Technical Specifications | | | | 10/02/2024 |
| 1.1.1 | Determine Project Scope | Team | 6 | 4 | |
| 1.1.2 | choose SDLC model | Team | 2 | 1 | |
| 1.1.3 | chooses the 2 API to used | Team | 1 | 1 | |
| 1.1.4 | Choose a GUI library | Team | 0.5 | 0.5 | |
| 1.1.5 | choose Programming Language | Team | 0.5 | 0.5 | |
| 1.1.6 | Determine Git Flow | Team | 2 | 1 | |
| 1.1.8 | Complete Technical Requirement | | 12 | 8 | |
| 1.2 | Develop Overall Model | | | | 10/02/2024 |
| 1.2.1 | Define Components | Team | 2 | 3 | |
| 1.2.2 | Define module | Team | 2 | 2 | |
| 1.2.3 | Define data flow | Team | 2 | 2 | |
| 1.2.4 | Define state management | Team | 2 | 2 | |
| 1.2.5 | Complete Develop Overall Model | | 8 | 9 | |
| 1.3 | Team Requirements | | | | 10/02/2024 |
| 1.3.1 | Assign Team Roles | Team | 0.5 | 0.5 | |
| 1.3.2 | Create Team Communication | Team | 0.5 | 0.5 | |
| 1.3.3 | Team meeting schedule | Team | 0.5 | 0.5 | |
| 1.3.4 | choose report method | Team | 2 | 0.5 | |
| 1.3.5 | create meeting report environment | Team | 1 | 0.5 | |
| 1.3.6 | Complete Team Requirement | | 4.5 | 4.5 | |
| 1.4 | Report milestone 1 | | | | 20/02/2024 |
| 1.4.1 | Overview of the project | Ravinesh | 0.5 | 0.5 | |
| 1.4.2 | Outline SDLC model chosen | Ravinesh | 0.5 | 0.5 | |
| 1.4.3 | Providing User stories | Ravinesh | 1 | 0.5 | |
| 1.4.4 | Providing persona | Ravinesh | 1 | 1 | |
| 1.4.5 | Outline technology stack selected | Ravinesh | 0.5 | 0.5 | |
| 1.4.6 | Outline API selected | Ravinesh | 0.5 | 0.5 | |
| 1.4.7 | Outline feature selected per API | Ravinesh | 0.5 | 0.5 | |
| 1.4.8 | providing project schedule (WBS + timeline) | Xavier | 5 | 6 | |
| 1.4.9 | Providing wireframe | Yujie | 5 | 7 | |
| 1.4.10 | Providing DFD level 0 and level 1 | Riley | 4 | 3 | |
| 1.4.11 | Providing swim lane diagram | Riley | 1 | 1 | |
| 1.4.12 | proofreading | team | 3 | 5 | |
| 1.4.12 | Complete Report milestone 1 | | 22.5 | 26 | |
| 1.5 | Presentation milestone 1 | | | | 19/02/2024 |
| 1.5.1 | Overview of the project | Ravinesh | 0.5 | 0.5 | |
| 1.5.2 | Outline SDLC model chosen | Xavier | 0.5 | 0.5 | |
| 1.5.3 | Outline technology stack selected | Ravinesh | 0.5 | 0.5 | |
| 1.5.4 | Outline API selected | Ravinesh | 0.5 | 0.5 | |
| 1.5.5 | Overview of the features chosen | Ravinesh | 0.5 | 0.5 | |
| 1.5.6 | overview of the wireframe and prototype | Yujie | 0.5 | 0.5 | |
| 1.5.7 | outline DFD level 0 | Riley | 0.5 | 0.5 | |
| 1.5.8 | Complete presentation milestone 1 | | 3.5 | 3.5 | |
| 1.6 | General Requirement | | 50.5 | 51 | |

Wireframes

| Homepage | Definition | Practise | Feedback |
|--|---|---|--|
|  <p>For searching English words and showing introductions to each feature</p> |  <p>Providing detailed definitions and visual explanations</p> |  <p>Allowing user to practise English writing with AI-generated image</p> |  <p>Providing feedback to help user improve</p> |

Prototype

[Link to Figma](#)



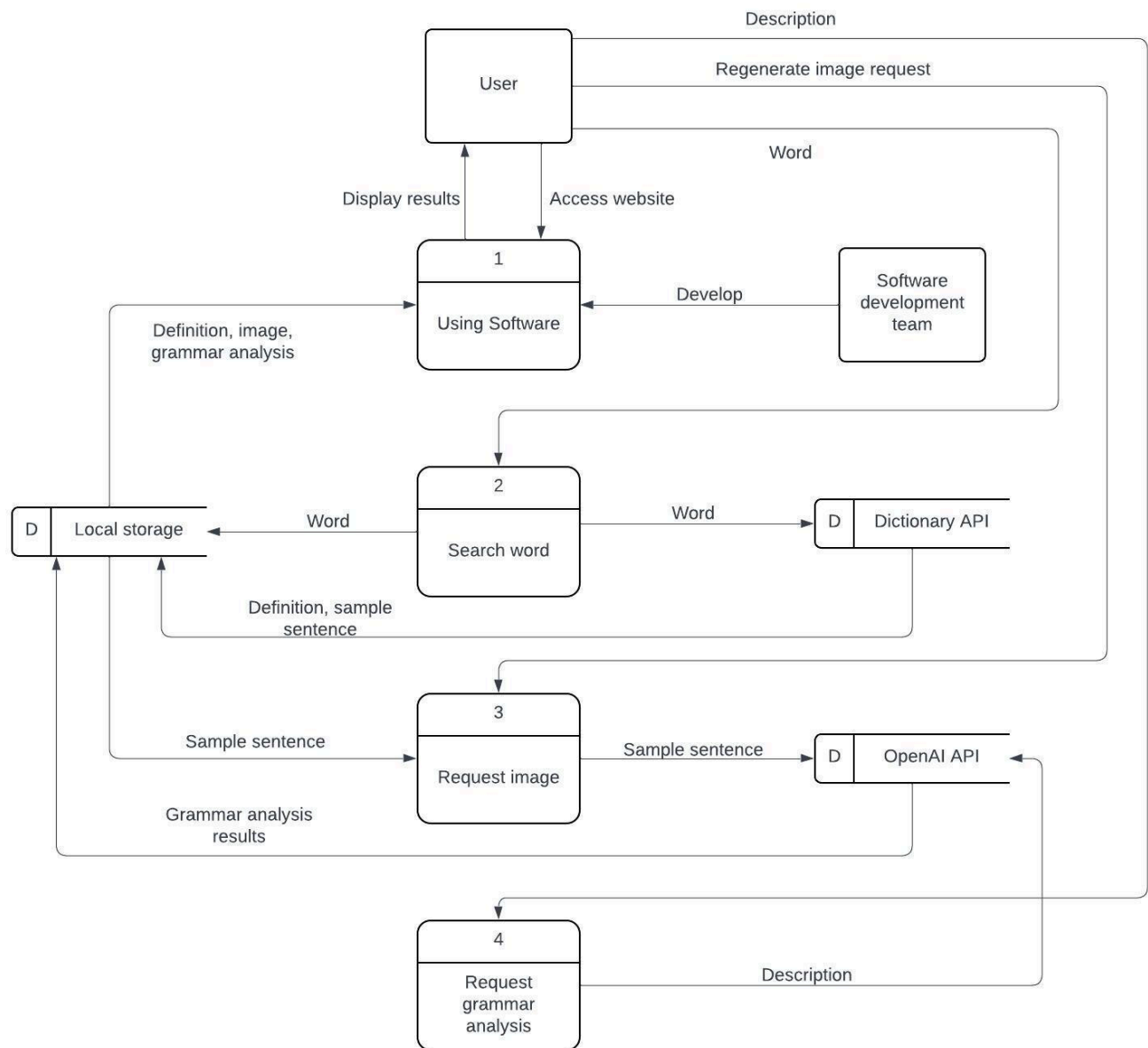
Data Flow Diagram

Level -0



- User uses the software and gives feedbacks
- Users receive images and information generated by Free Dictionary API and OpenAI API.
- The development team build and modify the software
- The development team receives feedbacks from users through the software

Level-1



Entities:

- Users
- Development team

Processes:

- Using the software
- Searching a word
- Request an image
- Request grammar analysis


Data stores:

- Free Dictionary API
- Open AI API
- Local storage on the user's local machine

Reference

- Free Dictionary API <https://github.com/meetDeveloper/freeDictionaryAPI>
- OpenAI API Image generation
<https://platform.openai.com/docs/guides/images?context=node>
- OpenAI API Text generation <https://platform.openai.com/docs/guides/text-generation>
- Feature-Driven Development
<https://www.bobstanke.com/blog/feature-driven-development>
- React: <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>
- Netlify:
<https://agilitycms.com/resources/posts/what-is-netlify-and-why-should-you-care-as-an-editor>
- Image: Cover Page <https://pixabay.com/vectors/idea-visualization-line-art-3976295/>
- Image: Project Overview
<https://pixabay.com/illustrations/project-management-project-planning-7446587/>
- Image: project SDLC
<https://pixabay.com/illustrations/project-plan-planning-quality-1987219/>
- Image technology stack
<https://pixabay.com/illustrations/banner-header-binary-zero-one-904887/>

Related Links

- GitHub Repository <https://github.com/RileyC9/EasyGrammar>
- Presentation Video  CPSC2350_Project_Presentation_1.mp4
- Prototype

<https://www.figma.com/file/Hb44ygMrI3tU2JhOfvViM6/CPSC2350-Project?type=design&node-id=0%3A1&mode=design&t=JQ7xgdhKH9eU3clo-1>

Appendix:

Communication

We are going to mainly stay in Discord. The communication server will be divided into general, case stories and ideas, different features, and documentation channels. There will also be a voice channel for team members to share their screen and discuss vocally.

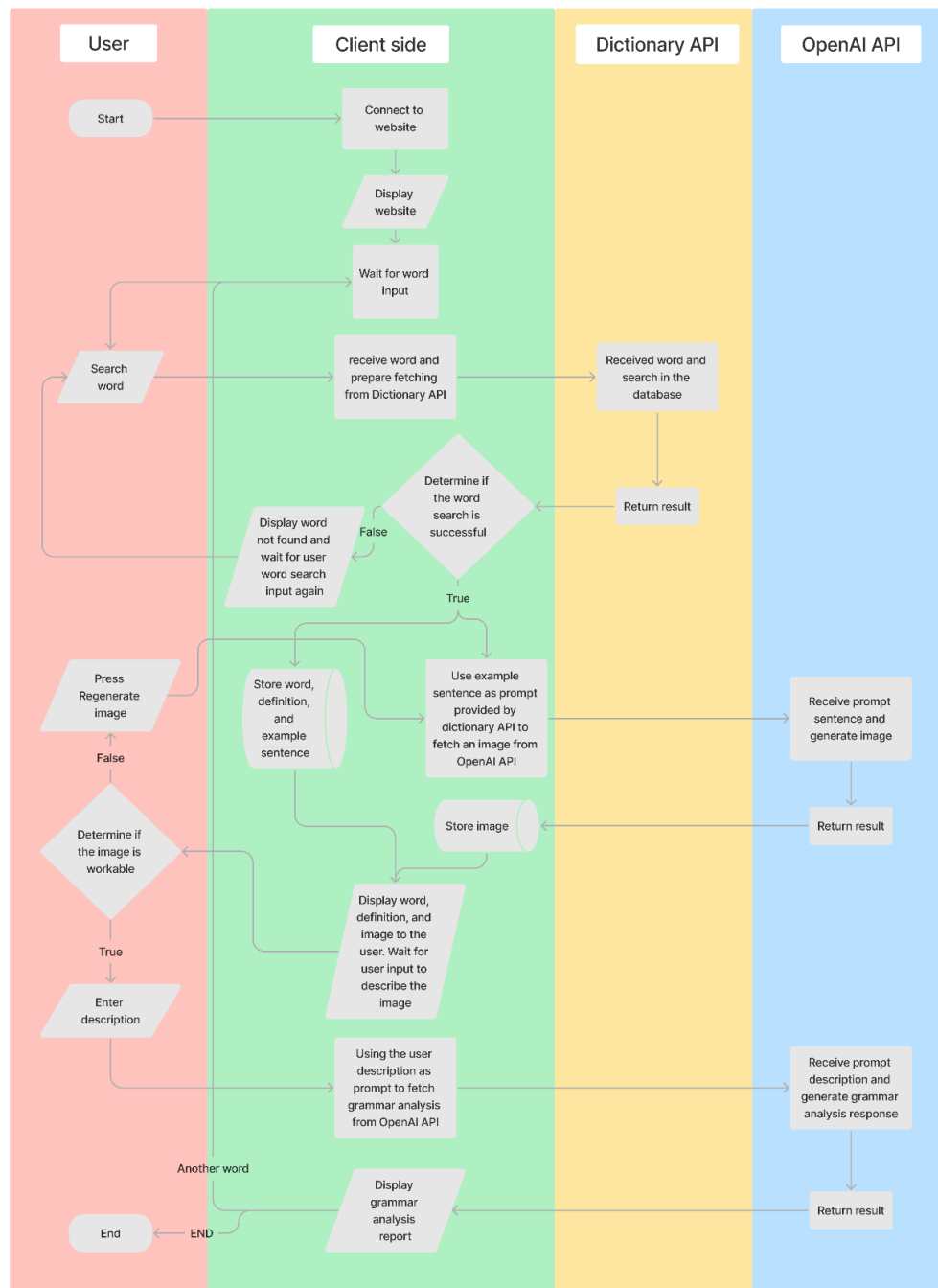
For the presentations, we will be using Zoom because of its record feature and reliable connection.

However, we may utilize the issues and project commenting features, dividing some communication loads (or important topics need to be recorded) from our Discord server. This decision will be made after the team member tries if it is necessary and the efficiency of such a communication channel.

More on Git branch management

Since we will be using FDD SDLC, we are going to create branches based on the features we are implementing. Everyone will be working on these branches until the feature is completed. Then we will merge it into the main branch.

Swimlane diagram



WBS

View  Project Schedule for latest version

| Project Schedule : Easy Grammar | | | | | |
|---------------------------------|-----------------------------------|-------------|-----------------|--------------|------------------------------|
| WEEK OF | | | | | |
| 2/20/2024 | | | | | |
| Task # | Task | Assigned To | Estimated Hours | Actual Hours | Comments/ Estimated complete |
| 1 | General Requirement | | | | |
| 1.1 | Technical Specifications | | | | 10/2/2024 |
| 1.1.1 | Determine Project Scope | Team | 6 | 4 | |
| 1.1.2 | choose SDLC model | Team | 2 | 1 | |
| 1.1.3 | chooses the 2 API to used | Team | 1 | 1 | |
| 1.1.4 | Choose a GUI library | Team | 0.5 | 0.5 | |
| 1.1.5 | choose Programming Language | Team | 0.5 | 0.5 | |
| 1.1.6 | Determine Git Flow | Team | 2 | 1 | |
| 1.1.8 | Complete Technical Requirement | | 12 | 8 | |
| 1.2 | Develop Overall Model | | | | 10/2/2024 |
| 1.2.1 | Define Components | Team | 2 | 3 | |
| 1.2.2 | Define module | Team | 2 | 2 | |
| 1.2.3 | Define data flow | Team | 2 | 2 | |
| 1.2.4 | Define state management | Team | 2 | 2 | |
| 1.2.5 | Complete Develop Overall Model | | 8 | 9 | |
| 1.3 | Team Requirements | | | | 10/2/2024 |
| 1.3.1 | Assign Team Roles | Team | 0.5 | 0.5 | |
| 1.3.2 | Create Team Communication | Team | 0.5 | 0.5 | |
| 1.3.3 | Team meeting schedule | Team | 0.5 | 0.5 | |
| 1.3.4 | choose report method | Team | 2 | 0.5 | |
| 1.3.5 | create meeting report environment | Team | 1 | 0.5 | |
| 1.3.6 | Complete Team Requirement | | 4.5 | 4.5 | |
| 1.4 | Report milestone 1 | | | | 20/2/2024 |
| 1.4.1 | Overview of the project | Ravinesh | 0.5 | 0.5 | |
| 1.4.2 | Outline SDLC model chosen | Ravinesh | 0.5 | 0.5 | |
| 1.4.3 | Providing User stories | Ravinesh | 1 | 0.5 | |
| 1.4.4 | Providing persona | Ravinesh | 1 | 1 | |

| | | | | | |
|--------|---|----------|------|-----|-----------|
| 1.4.5 | Outline technology stack selected | Ravinesh | 0.5 | 0.5 | |
| 1.4.6 | Outline API selected | Ravinesh | 0.5 | 0.5 | |
| 1.4.7 | Outline feature selected per API | Ravinesh | 0.5 | 0.5 | |
| 1.4.8 | providing project schedule (WBS + timeline) | Xavier | 5 | 6 | |
| 1.4.9 | Providing wireframe | Yujie | 5 | 7 | |
| 1.4.10 | Providing DFD level 0 and level 1 | Riley | 4 | 3 | |
| 1.4.11 | Providing swim lane diagram | Riley | 1 | 1 | |
| 1.4.12 | proofreading | team | 3 | 5 | |
| 1.4.12 | Complete Report milestone 1 | | 22.5 | 26 | |
| 1.5 | Presentation milestone 1 | | | | 19/2/2024 |
| 1.5.1 | Overview of the project | Ravinesh | 0.5 | 0.5 | |
| 1.5.2 | Outline SDLC model chosen | Xavier | 0.5 | 0.5 | |
| 1.5.3 | Outline technology stack selected | Ravinesh | 0.5 | 0.5 | |
| 1.5.4 | Outline API selected | Ravinesh | 0.5 | 0.5 | |
| 1.5.5 | Overview of the features chosen | Ravinesh | 0.5 | 0.5 | |
| 1.5.6 | overview of the wireframe and prototype | Yujie | 0.5 | 0.5 | |
| 1.5.7 | outline DFD level 0 | Riley | 0.5 | 0.5 | |
| 1.5.9 | Complete presentation milestone 1 | | 3.5 | 3.5 | |
| 1.6 | General Requirement | | 50.5 | 51 | |
| | | | | | |
| 2 | GUI | | | | |
| 2.1 | Layout | | | | 20/2/2024 |
| 2.1.1 | Paper draft layout | Yujie | 2 | | |
| 2.1.2 | computer Draf layout | Yujie | 4 | | |
| 2.1.3 | Complete Layout | | 6 | | |
| 2.2 | Graphics | | | | 2/3/2024 |
| 2.2.1 | Create a new Background | Yujie | 1 | | |
| 2.2.2 | searching pictures related to education | Yujie | 1 | | |
| 2.2.3 | Photoshop pictures | Yujie | 1 | | |
| 2.2.4 | Add New Font | Yujie | 0.5 | | |
| 2.2.5 | Select Typo | Yujie | 0.5 | | |
| 2.2.6 | Complete Graphics | | 4 | | |
| 2.3 | Resizing | | | | 23/3/2024 |
| 2.3.1 | research about standard | Yujie | 2 | | |
| 2.3.2 | Determine Limitations | Yujie | 1 | | |

| | | | | | |
|--------|---|--------------|-----|--|-----------|
| 2.3.3 | Implement Resizing | Yujie | 3 | | |
| 2.3.4 | Complete Resizing | | 5 | | |
| 2.3.5 | contingency time | Yujie/Team | 2 | | |
| 2.4 | complete GUI | | 21 | | |
| | | | | | |
| 3 | Core | | | | |
| 3.1 | set up the environment | | | | 29/2/2024 |
| 3.1.1 | Set up React | Riley | 1 | | |
| 3.1.2 | Set up Express | Riley | 1 | | |
| 3.1.3 | set up node module | Riley | 1 | | |
| 3.1.4 | create folders | Riley | 0.5 | | |
| 3.1.5 | create components | Riley | 0.5 | | |
| 3.1.6 | complete React Set Up | | 5 | | |
| 3.2 | API selection | | | | 10/2/2024 |
| 3.2.1 | research APIs options | Team | 4 | | |
| 3.2.2 | Select APIs | Team | 1 | | |
| 3.2.3 | Complete API selection | | 5 | | |
| 3.3 | Feature word validation | | | | 6/3/2024 |
| 3.3.1 | research about the feature | Riley/Xavier | 2 | | |
| 3.3.2 | set up feature environment | Riley/Xavier | 1 | | |
| 3.3.3 | prototype | Riley/Xavier | 6 | | |
| 3.3.4 | feedback from the prototype | Team | 1 | | |
| 3.3.5 | improvement/modification of the prototype | Team | 2 | | |
| 3.3.6 | Unit Testing | Team | 3 | | |
| 3.3.7 | integrate word validation to GUI | Riley/Xavier | 2 | | |
| 3.3.8 | error handling | Riley/Xavier | 2 | | |
| 3.3.9 | adapting the feature with the GUI | Yujie | 2 | | |
| 3.3.10 | Integration Testing | Team | 3 | | |
| 3.3.11 | retrospective meeting | Team | 1 | | |
| 3.3.12 | create feature documentation | Ravinesh | 1 | | |
| 3.3.13 | contingency time | Team | 2 | | |
| 3.3.14 | Complete Feature word validation | | 28 | | |
| 3.4 | Feature AI text to image generation | | | | 10/3/2024 |
| 3.4.1 | Research about the feature | Riley/Xavier | 2 | | |
| 3.4.2 | Set up feature environment | Riley/Xavier | 1 | | |
| 3.4.3 | Prototype | Riley/Xavier | 6 | | |
| 3.4.4 | Feedback from the prototype | Team | 1 | | |

| | | | | | |
|--------|--|--------------|----|--|-----------|
| 3.4.5 | Improvement/modification of the prototype | Team | 2 | | |
| 3.4.6 | Unit Testing | Team | 3 | | |
| 3.4.7 | Integrate Ai text-image to GUI | Riley/Xavier | 2 | | |
| 3.4.8 | error handling | Riley/Xavier | 2 | | |
| 3.4.9 | adapting the feature to the GUI | Yujie | 2 | | |
| 3.4.10 | Integration Testing | Team | 3 | | |
| 3.4.11 | retrospective meeting | Team | 1 | | |
| 3.4.12 | create feature documentation | Ravinesh | 1 | | |
| 3.4.13 | contingency time | Team | 2 | | |
| 3.4.14 | Complete Feature AI text to image generation | | 28 | | |
| 3.5 | Feature Identify grammar mistakes | | | | 16/3/2024 |
| 3.5.1 | research about the feature | Riley/Xavier | 2 | | |
| 3.5.2 | set up feature environment | Riley/Xavier | 1 | | |
| 3.5.3 | prototype | Riley/Xavier | 6 | | |
| 3.5.4 | feedback from the prototype | Team | 1 | | |
| 3.5.5 | improvement/modification of the prototype | Team | 2 | | |
| 3.5.6 | Unit Testing | Team | 3 | | |
| 3.5.7 | integrate identify grammar mistakes to GUI | Riley/Xavier | 2 | | |
| 3.5.8 | error handling | Riley/Xavier | 2 | | |
| 3.5.9 | adapting the feature to the GUI | Yujie | 2 | | |
| 3.5.10 | Integration Testing | Team | 3 | | |
| 3.5.11 | retrospective meeting | Team | 1 | | |
| 3.5.12 | create feature documentation | Ravinesh | 1 | | |
| 3.5.13 | contingency time | Team | 2 | | |
| 3.5.14 | Complete feature identify grammar mistakes | | 28 | | |
| 3.6 | Contextual sentence | | | | 20/3/2024 |
| 3.6.1 | research about the feature | Riley/Xavier | 2 | | |
| 3.6.2 | set up feature environment | Riley/Xavier | 1 | | |
| 3.6.3 | prototype | Riley/Xavier | 6 | | |
| 3.6.4 | feedback from the prototype | Team | 1 | | |
| 3.6.5 | improvement/modification of the prototype | Team | 2 | | |
| 3.6.6 | Unit Testing | Team | 3 | | |
| 3.6.7 | integrate contextual sentence to GUI | Riley/Xavier | 2 | | |
| 3.6.8 | error handling | Riley/Xavier | 2 | | |

| | | | | | |
|--------|---|--------------|----|--|-----------|
| 3.6.9 | adapting the feature to the GUI | Yujie | 2 | | |
| 3.6.10 | Integration Testing | Team | 3 | | |
| 3.6.11 | retrospective meeting | Team | 1 | | |
| 3.6.12 | create feature documentation | Ravinesh | 1 | | |
| 3.6.13 | contingency time | Team | 2 | | |
| 3.6.14 | complete Contextual sentence | | 28 | | |
| 3.7 | Feature word definition | | | | 9/3/2024 |
| 3.7.1 | research about the feature | Riley/Xavier | 2 | | |
| 3.7.2 | set up feature environment | Riley/Xavier | 1 | | |
| 3.7.3 | prototype | Riley/Xavier | 6 | | |
| 3.7.4 | feedback from the prototype | Team | 1 | | |
| 3.7.5 | improvement/modification of the prototype | Team | 2 | | |
| 3.7.6 | Unit Testing | Team | 3 | | |
| 3.7.7 | integrate word definition to GUI | Riley/Xavier | 2 | | |
| 3.7.8 | error handling | Riley/Xavier | 2 | | |
| 3.7.9 | adapting the feature to the GUI | Yujie | 2 | | |
| 3.7.10 | Integration Testing | Team | 3 | | |
| 3.7.11 | retrospective meeting | Team | 1 | | |
| 3.7.12 | create feature documentation | Ravinesh | 1 | | |
| 3.7.13 | contingency time | Team | 2 | | |
| 3.7.14 | complete Feature word definition | | 28 | | |
| 3.8 | Feature Evaluation of accuracy | | | | 20/3/2024 |
| 3.8.1 | research about the feature | Riley/Xavier | 2 | | |
| 3.8.2 | set up feature environment | Riley/Xavier | 1 | | |
| 3.8.3 | prototype | Riley/Xavier | 6 | | |
| 3.8.4 | feedback from the prototype | Team | 1 | | |
| 3.8.5 | improvement/modification of the prototype | Team | 2 | | |
| 3.8.6 | Unit Testing | Team | 3 | | |
| 3.8.7 | integrate evaluation of accuracy to GUI | Riley/Xavier | 2 | | |
| 3.8.8 | error handling | Riley/Xavier | 2 | | |
| 3.8.9 | adapting the feature to the GUI | Yujie | 2 | | |
| 3.8.10 | Integration Testing | Team | 3 | | |
| 3.8.11 | retrospective meeting | Team | 1 | | |
| 3.8.12 | create feature documentation | Ravinesh | 1 | | |
| 3.8.13 | contingency time | Team | 2 | | |

| | | | | | |
|--------|---|----------|-----|--|-----------|
| 3.8.14 | complete Feature Evaluation of accuracy | | 28 | | |
| 3.9 | Testing | | | | 23/3/2024 |
| 3.9.2 | Specific Questions | Team | 2 | | |
| 3.9.3 | Corner cases | Team | 2 | | |
| 3.9.4 | System Testing | Team | 5 | | |
| 3.9.5 | Debugging | Team | 2 | | |
| 3.9.6 | contingency time | Team | 2 | | |
| 3.9.7 | Complete Testing | | 13 | | |
| 3.10 | Complete core | | 191 | | |
| | | | | | |
| 4 | Documentations | | | | |
| 4.1 | Initiation phases | | | | 20/3/2024 |
| 4.1.1 | Brainstorm Ideas and answer | Ravinesh | 2 | | |
| 4.1.2 | ideas that were not selected and why | Ravinesh | 2 | | |
| 4.1.3 | Create Document and Report | Ravinesh | 1 | | |
| 4.1.4 | Complete initiation phase | | 5 | | |
| 4.2 | Final presentation | | | | |
| 4.2.1 | Advertisement video | Yujie | 4 | | |
| 4.2.2 | Overview of the project | Ravinesh | 1 | | |
| 4.2.3 | feature details | Xavier | 1 | | |
| 4.2.4 | issue encounter | Riley | 1 | | |
| 4.2.5 | test details | xavier | 1 | | |
| 4.2.6 | GUI details | Yujie | 1 | | |
| 4.2.7 | real-time test | Riley | 1 | | |
| 4.2.8 | Complete Final Presentation | | 9 | | |
| 4.2 | Project report | | | | 2/4/2024 |
| 4.2.1 | Project description | Ravinesh | 2 | | |
| 4.2.2 | providing project schedule (WBS + timeline) | Xavier | 5 | | |
| 4.2.3 | Project limitation | Ravinesh | 3 | | |
| 4.2.4 | software Prototype | Ravinesh | 3 | | |
| 4.2.5 | User documentation | Ravinesh | 3 | | |
| 4.2.5 | testing documentation | Ravinesh | 4 | | |
| 4.2.6 | Code documentation | Ravinesh | 5 | | |
| 4.2.7 | Meeting Documentation | Ravinesh | 2 | | |
| 4.2.8 | proofreading | Team | 2 | | |
| 4.2.9 | Complete Project Report | | 29 | | |
| 4.3 | Complete Documentations | | 34 | | |