Pokémon ID Application for Enthusiasts

by Jocelyn Dzuong, Bryan Bravo, Roberto Hernandez, Cesar Santiago

Table of Contents

Introduction

Description of the customer/setting for the project

- **Salient characteristics of the customer or sponsoring organization**
    - Fans of Pokémon from children to adults
    - Children and adults who are enthusiastic about learning to identify more Pokémon.
- **Description of the salient characteristics of the application/tool**
    - **What is the background of the project idea? What is the problem?**
        - We wish to gamify learning about Pokémon. The application is initially intended as a tool for the game's enthusiasts or people wanting to learn more about each creature. The user will snap a picture of a Pokémon product that they may come across, an AI system will respond with information about that Pokémon and give the user a score on how good the image may be. This score is used in leaderboards, so each user competes with each other to snap the best pictures of Pokémon they find. The system will also give them back a trading card for each of the Pokémon they have collected.

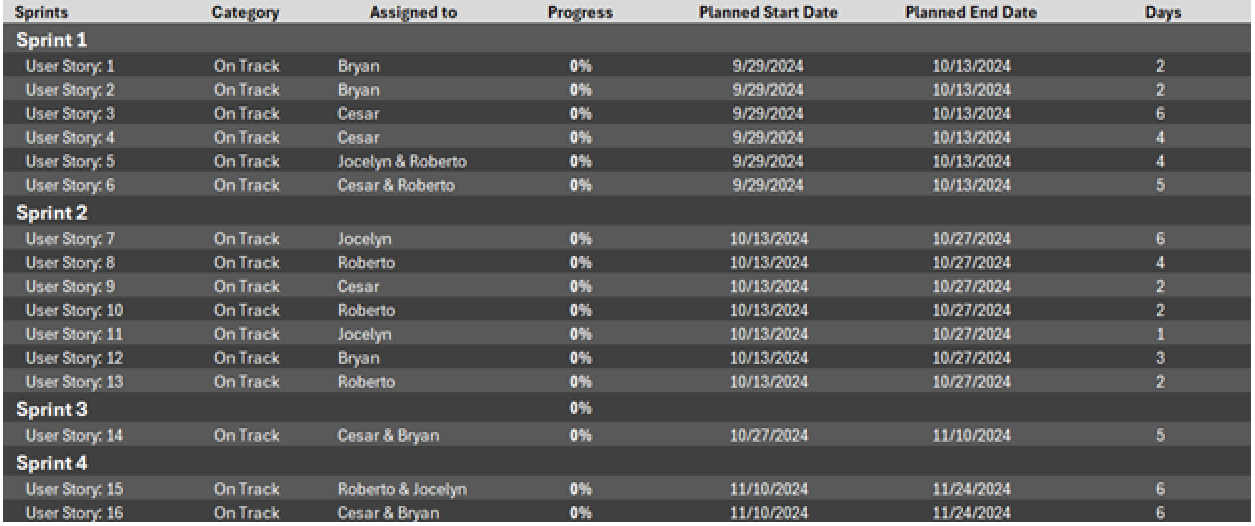
    - **How is your application/tool different from what already exists?**
        - Existing Pokémon products focus primarily in a game world, and Pokémon are confined to it. Everywhere in the world a person can come across a Pokémon product, which they often overlook because they aren't already attached to the characters. This app can help users recognize these products and attach them to the creatures that they represent. While also giving the user props for finding more Pokémon and snapping better images of these creatures.

    - **What are the implications of the tool you develop?**
        - This tool can be very helpful for people that wish to get into the franchise that they see so much in stores, outlets, and advertisement, but are too overwhelmed by the number of creatures

or their distaste of the medium of video games. Now, these people can better communicate with others about these creatures, without spending hundreds of hours in a virtual world. The tool can also be beneficial to the Pokémon company by introducing a more accessible way of navigating these products.

- **Other important contextual issues especially any external constraints placed on the application/tool.**
  - While the app aims to make Pokémon products more accessible through image recognition, several external constraints and factors need to be considered:
    - Intellectual Property (IP): Using Pokémon images and branding will require licensing from The Pokémon Company, which may limit content unless permissions are secured.
    - Privacy and Security: The app will handle user photos, so it must comply with privacy regulations to protect personal data and images.
    - AI Accuracy: Ensuring the AI can reliably recognize Pokémon products in various conditions, such as different lighting or angles, presents a technical challenge.
    - Device Compatibility: The app must work across various devices, from high-end to older models, which could affect image recognition performance on some platforms.
    - Localization: Regional variations in Pokémon products and language support need to be addressed for global accessibility.
    - Additionally, targeting both children and adults requires balancing simplicity with engagement, while the leaderboards and monetization need careful design to avoid negative experiences like cheating or intrusive ads, especially for younger users.

- **Outline a proposed schedule for your project**



Sprints	Category	Assigned to	Progress	Planned Start Date	Planned End Date	Days
Sprint 1						
User Story: 1	On Track	Bryan	0%	9/29/2024	10/13/2024	2
User Story: 2	On Track	Bryan	0%	9/29/2024	10/13/2024	2
User Story: 3	On Track	Cesar	0%	9/29/2024	10/13/2024	6
User Story: 4	On Track	Cesar	0%	9/29/2024	10/13/2024	4
User Story: 5	On Track	Jocelyn & Roberto	0%	9/29/2024	10/13/2024	4
User Story: 6	On Track	Cesar & Roberto	0%	9/29/2024	10/13/2024	5
Sprint 2						
User Story: 7	On Track	Jocelyn	0%	10/13/2024	10/27/2024	6
User Story: 8	On Track	Roberto	0%	10/13/2024	10/27/2024	4
User Story: 9	On Track	Cesar	0%	10/13/2024	10/27/2024	2
User Story: 10	On Track	Roberto	0%	10/13/2024	10/27/2024	2
User Story: 11	On Track	Jocelyn	0%	10/13/2024	10/27/2024	1
User Story: 12	On Track	Bryan	0%	10/13/2024	10/27/2024	3
User Story: 13	On Track	Roberto	0%	10/13/2024	10/27/2024	2
Sprint 3			0%			
User Story: 14	On Track	Cesar & Bryan	0%	10/27/2024	11/10/2024	5
Sprint 4						
User Story: 15	On Track	Roberto & Jocelyn	0%	11/10/2024	11/24/2024	6
User Story: 16	On Track	Cesar & Bryan	0%	11/10/2024	11/24/2024	6

  - **Show your User Story backlog**
    - ■ [US:1] "As a Developer, I want to build/create a database with the purpose to be able to store information of users.
      - Acceptance Criteria: Verify that a database has been created allowing for data to be stored."
    - ■ [US:2] "As a user, I want to create an account using my email and 2FA so I can securely access my Pokedex.
      - Acceptance Criteria: Verify that the user can create an account and login to their dashboard successfully."
    - ■ [US:3] "As a developer, I want to integrate a PyTorch-based convolutional neural network model trained on 1000+ Pokémon so that the app can accurately recognize scanned images with the model.
      - Acceptance Criteria: Verify that the PyTorch model is trained upon 1000+ Pokémon images and that the app is able to interface with the model in the integrated camera."
    - ■ [US:4] "As a developer, I want to build a system that requests external information from the public PokeAPI API with the purpose of serving the user the data of the Pokémon that they encountered.
      - Acceptance Criteria: Verify that the application can pull Pokémon data from the corresponding Pokémon from the PokeAPI."

- [US:5] "As a user, I want a colorful and intuitive interface so that I can navigate the app easily without confusion.
    - Acceptance Criteria: Verify that the user upon opening the Flutter app can login and navigate through the menu buttons and UI seamlessly with no tutorial needed."
- [US:6] "As a user, I want to use my device’s camera to scan Pokémon images I see in real life so that the app can identify the Pokémon for me
    - Acceptance Criteria: Verify that when the user utilizes their phone camera in the app that the camera can identify a Pokémon image by a bounded box"
- [US:7] "As a developer, I want to build a user interface (UI) that neatly displays the Pokémon information requested from the external PokeAPI, with the purpose of serving the user a better experience.
    - Acceptance Criteria: Verify that a front-end system is developed that displays the contents of the request in a desired manner."
- [US:8] "As a user, I want to access an account, so that I may view the contents of the account and utilize the app’s features.
    - Acceptance Criteria: Verify that a user is able to log into an existing account."
- [US:9] "As a user, I want to delete an account while logged into the application, so that my information can be safely removed.
    - Acceptance Criteria: Verify that a user is able to delete their profile information from the database, and that it is no longer accessible."
- [US:10] "As an Administrator, I want the privilege of changing a user’s username with the purpose of moderating malicious user interaction.
    - Acceptance Criteria: In administrator mode, allow the ability to change another user’s username."

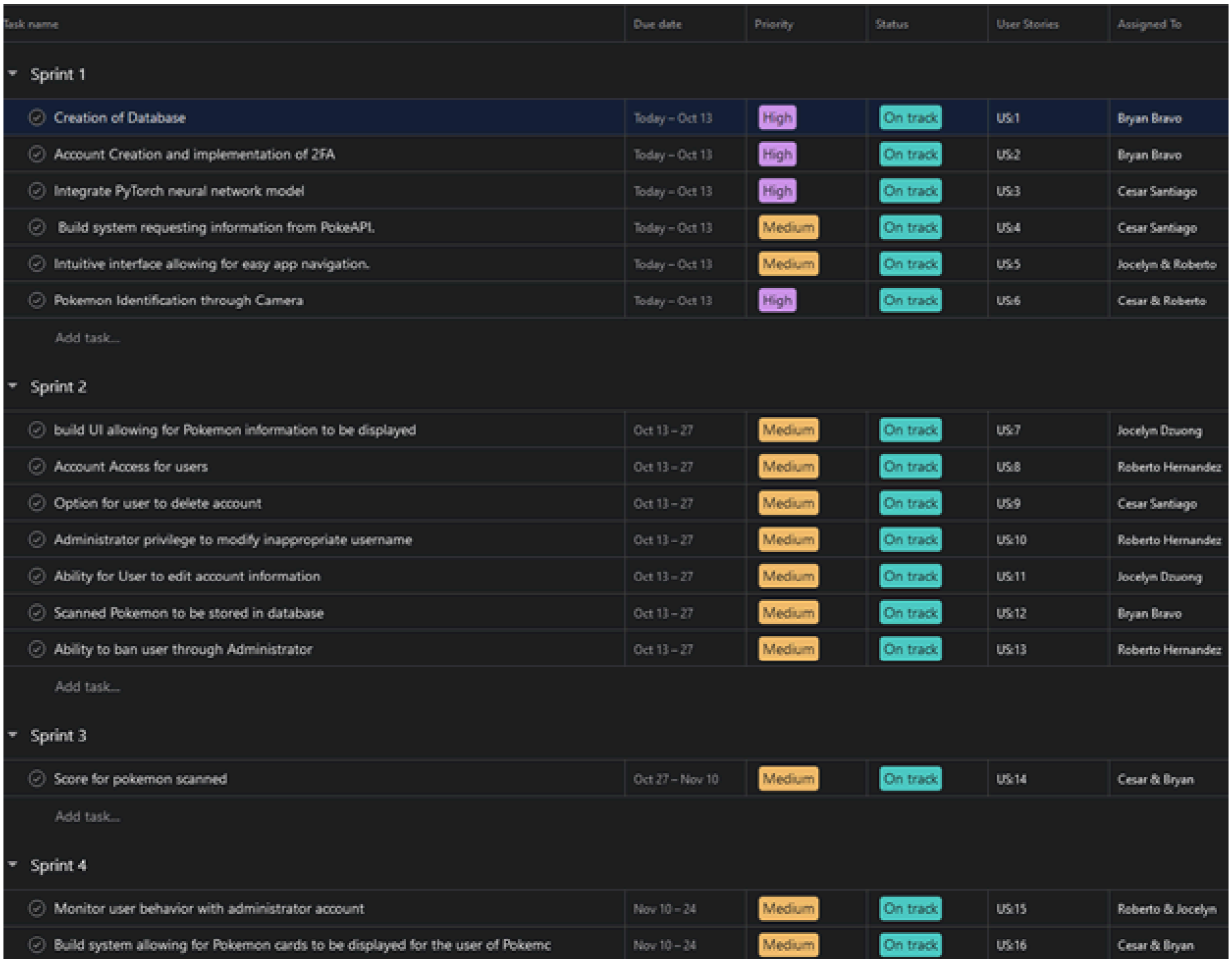
- [US:11] "As a user, I want to edit my account information while logged into the application, so that I can modify existing profile information.
    - Acceptance Criteria: Verify that a user is able to edit all of the profile information, such as username, email, and password securely."
- [US:12] "As a user, I want to save my scanned Pokémon entries to a database so that I can access them easily with a wireless connection.
    - Acceptance Criteria: Verify that the user can access their scanned Pokémon entries in the app via a Firebase backend instance."
- [US:13] "As an Administrator, I want the privilege of banning a user from the system, with the purpose of moderating misbehaving users.
    - Acceptance Criteria: In an administrator account, be able to ban a user through their username."
- [US:14] "As a developer, I want to give the user a score for the image they snapped, with the purpose of giving the user incentive to create better images of Pokémon.
    - Acceptance Criteria: Verify that the trained model returns a score that resembles how close the image they took is from the Pokémon."
- [US:15] "As an Administrator, I want access to an account with administrative privileges, with the intention of being able to moderate user behavior.
    - Acceptance Criteria: Verify that a user can have administrator privileges."
- [US:16] "As a developer, I want to build a system that requests external information from the public Pokémon trading card game, with the purpose of displaying the TCG’s cards to the user when a Pokémon is collected by them.

        - Acceptance Criteria: Verify that a user can pull information from the Pokémon TCG API and access the cards they collected."

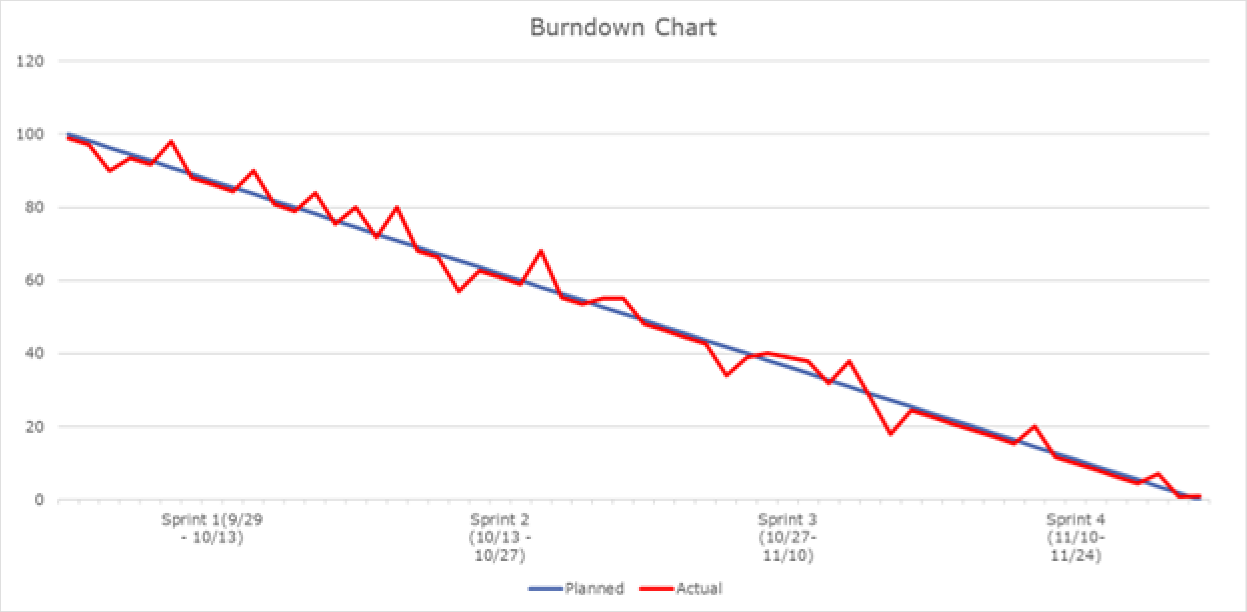
- **Describe the process used to determine user story allocation**
    - For each user story we analyzed the needs of one of our stakeholders, be it the users, admins, or developers. Then we isolated an issue that concerned them in the application. We then justified the issue, and why they would be interested in it. And then we created the requirement that needs to be completed to be accepted.

- **Determine the sprint period (1 – 3 weeks)**
    - 2 weeks

- Allocate the backlog to the Sprints



Task name	Due date	Priority	Status	User Stories	Assigned To
Sprint 1					
Creation of Database	Today – Oct 13	High	On track	US:1	Bryan Bravo
Account Creation and implementation of 2FA	Today – Oct 13	High	On track	US:2	Bryan Bravo
Integrate PyTorch neural network model	Today – Oct 13	High	On track	US:3	Cesar Santiago
Build system requesting information from PokeAPI.	Today – Oct 13	Medium	On track	US:4	Cesar Santiago
Intuitive interface allowing for easy app navigation.	Today – Oct 13	Medium	On track	US:5	Jocelyn & Roberto
Pokemon Identification through Camera	Today – Oct 13	High	On track	US:6	Cesar & Roberto
Add task...					
Sprint 2					
build UI allowing for Pokemon information to be displayed	Oct 13 – 27	Medium	On track	US:7	Jocelyn Dzuong
Account Access for users	Oct 13 – 27	Medium	On track	US:8	Roberto Hernandez
Option for user to delete account	Oct 13 – 27	Medium	On track	US:9	Cesar Santiago
Administrator privilege to modify inappropriate username	Oct 13 – 27	Medium	On track	US:10	Roberto Hernandez
Ability for User to edit account information	Oct 13 – 27	Medium	On track	US:11	Jocelyn Dzuong
Scanned Pokemon to be stored in database	Oct 13 – 27	Medium	On track	US:12	Bryan Bravo
Ability to ban user through Administrator	Oct 13 – 27	Medium	On track	US:13	Roberto Hernandez
Add task...					
Sprint 3					
Score for pokemon scanned	Oct 27 – Nov 10	Medium	On track	US:14	Cesar & Bryan
Add task...					
Sprint 4					
Monitor user behavior with administrator account	Nov 10 – 24	Medium	On track	US:15	Roberto & Jocelyn
Build system allowing for Pokemon cards to be displayed for the user of Pokemc	Nov 10 – 24	Medium	On track	US:16	Cesar & Bryan



- **Describe your team and the proposed roles and responsibilities for your teammates**
    - **Front End – Jocelyn Dzuong**
        - The role of the Front -End developer will be to work on contributing on designing and implementing a user-friendly, colorful and intuitive interface in a Flutter app. Which will ensure seamless navigation without the need for tutorials. The front-end developer will also work on building a front-end system to efficiently display data from external APIs (i.e. PokeAPI), which will aim to improve user experience by delivering the requested information in a clear and organized manner.
    - **Back End - Bryan Bravo**
        - The responsibilities pertaining to the role of back-end will be to specialize in the designing, implementation and maintenance of the database in which users' data would be stored and accessed when using the application. The responsibilities pertaining to the role of back-end will be to specialize in the designing, implementation and the maintenance of the database. Which will store and access users' data when using the application alongside key components that are essential for the project to display such as a user's Pokémon's that have been scanned and user information that pertains to accounts.
    - **Machine Learning – Cesar Santiago**
        - The role pertaining to Machine learning will be to develop and integrate a PyTorch based convolutional neural network (CNN) model trained on a multitude of Pokémon images for image recognition. It will also feature designing the app to interface with the model, enabling accurate recognition of scanned images via an integrated camera. Additionally, within the role it will work on implementing systems that pull external data from public API's (e.g

PokeAPI and Pokémon TCG API) to enhance the user experience by providing real-time Pokémon data and trading card information. They will also ensure user account management, featuring the ability to delete profile information from the database.

- **Project Manager/Admin - Roberto Hernandez**
    - For the role of Project Manager/admin, it will consist of managing user account functionality and administrative privileges within the app. Ensuring users can securely log into their accounts and access the app's features. As well as implementing administrator controls to moderate user interactions, including the ability to change usernames and ban users from the system. They will also ensure user account management, featuring the ability to delete profile information from the database while focusing on maintaining a safe and user-friendly environment by enforcing security and moderation protocols.

Design

What is Design?

Software design refers to the process of organizing the software's architecture, interfaces, and modules to satisfy specified requirements effectively from stakeholders (IEEE, 2024). It consists of selecting system elements in order to optimize aspects such as functionality, performance, and usability. Typically, detailed and sufficient software design supports accessible and smooth user experiences, providing a solid foundation to extend the software in terms of features and enhancements. Software design in mobile apps in this case considers user-centric design, accessibility, and responsiveness to incorporate effortless usability.

Design Process

Our team adopted an agile development process, combining principles of design with iterative feedback in order to develop our Pokémon detection app. We believe this approach allows us to remain responsive to user feedback, improve accessibility, and create an engaging gamified experience (Kaya & Ercag, 2023). Given the focus on machine learning crossed with strong yet simple aesthetics, we approached development emphasizing both technical and design-driven needs in mind:

1. The team began by understanding user needs, emphasizing accessibility, gamification, and aesthetic appeal. Since the app required real-time image recognition of Pokémon from real-world images, we incorporated machine learning for object detection and decided on Flutter and Dart to streamline cross-platform development.

2. We used Flutter for rapid UI prototyping to achieve smooth animations and an intuitive layout. Dart's language flexibility enabled us to efficiently implement the app's business logic while focusing on simple and accessible design. We focused on large, readable icons, simple color schemes, and high-contrast elements to maximize accessibility.

3. A core feature of the app involves using a machine learning model to detect and identify Pokémon from a live camera feed and/or user image input. To manage this, we trained a T5 model compatible with Flutter's TensorFlow Lite plugin to ensure seamless on-device inference as well as quick response times.

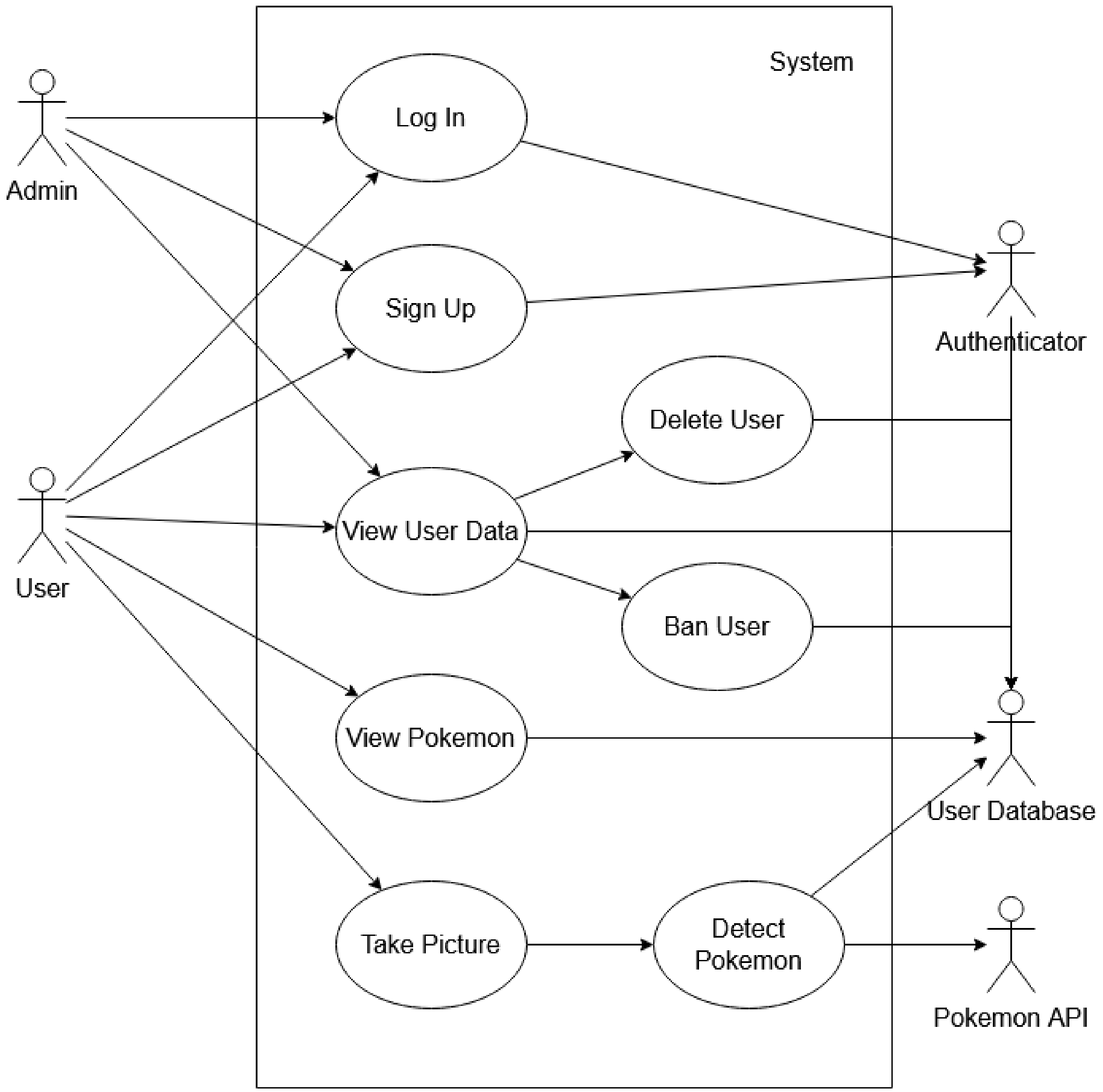
4. After prototyping ideas, we conducted bi-weekly sprints to review our progress. Each sprint included a feedback session where we tested the app and changes to identify areas for refinement. Adjustments were made to maintain both aesthetic quality and accessibility such as modifying elements for readability and ensuring ease of interaction for users of all abilities.

5. Firebase was chosen for its real-time database support for the scoreboard feature. Firebase's ability to integrate with Flutter allowed us to manage user data and ensure the scalability needed to support a large number of users. With real-time updates, users can see their progress and compete against others.

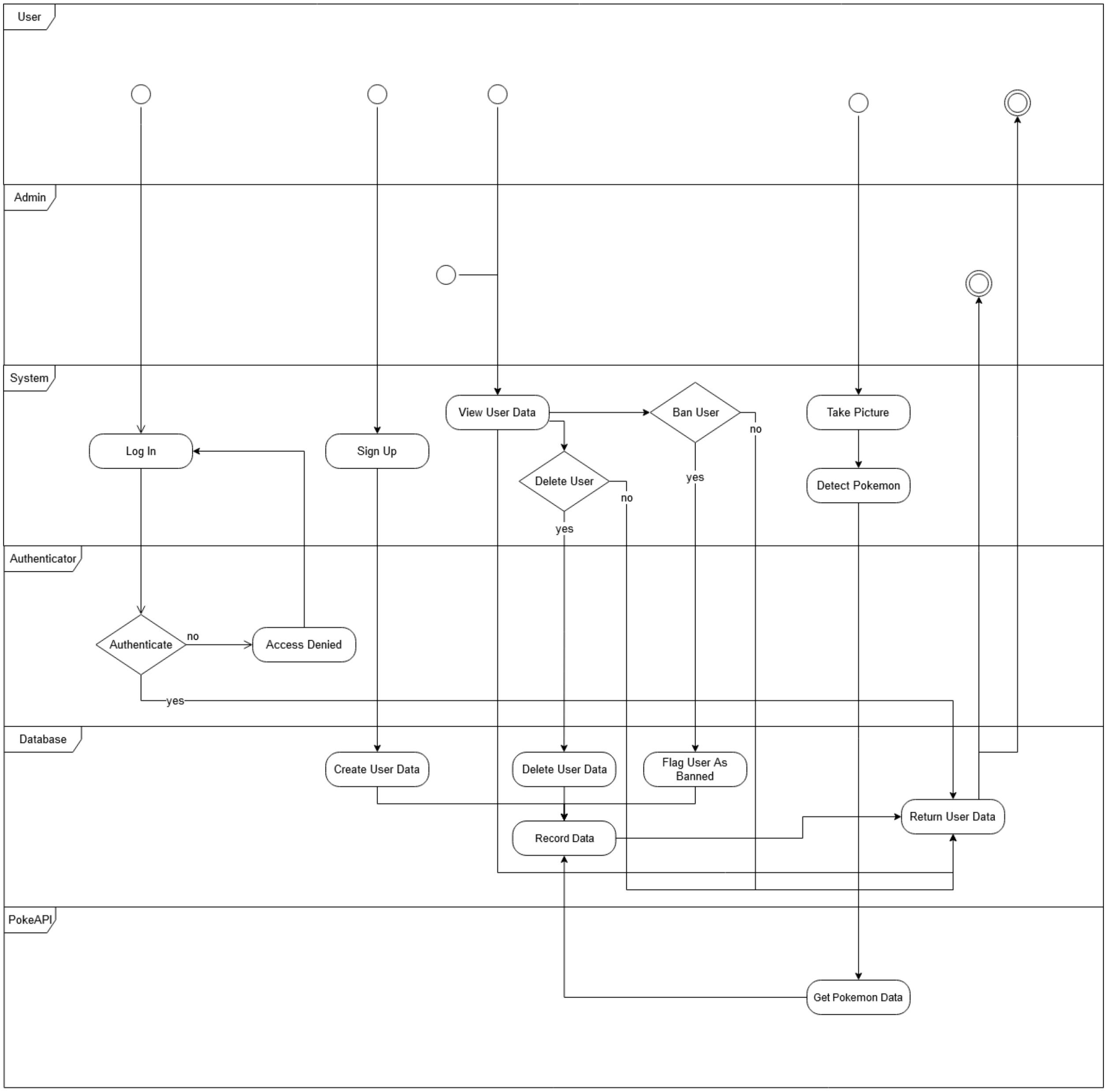
6. We prioritized color contrast, large touch targets, and clear text hierarchies, ensuring usability for all users, including those with visual or motor impairments. Visual consistency and intuitive navigation were key, as well as carefully selected animations to create a smooth, engaging user experience without overwhelming users.

Diagrams and Mockups

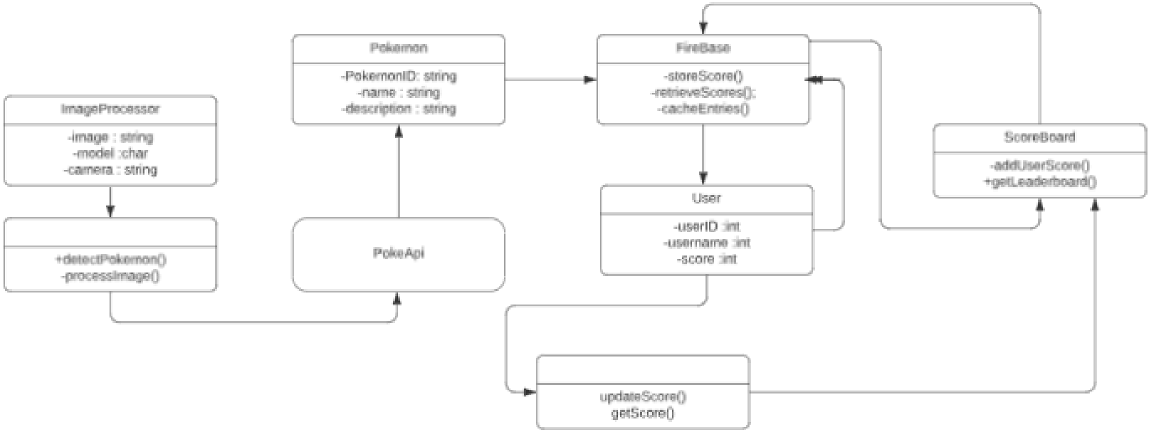
1. Use case Diagrams



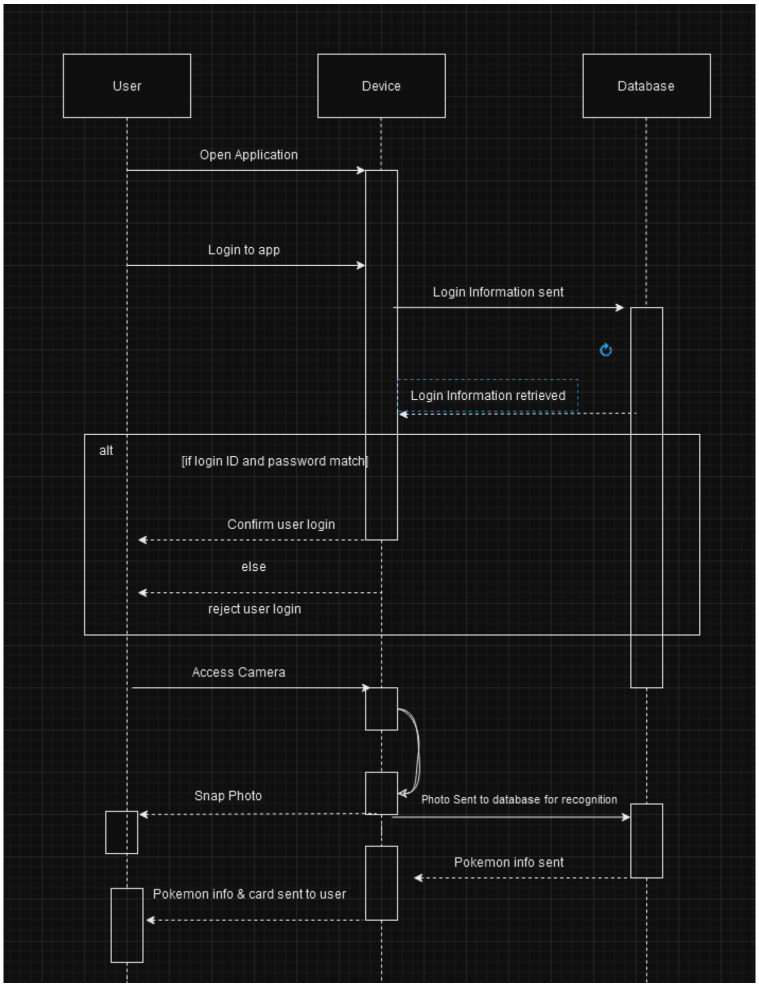
2. Activity Diagrams (if applicable)



3. Class Diagrams



4. Sequence diagrams



Rationale Management

1. The Issues That Were Addressed
    - Accessibility: Ensuring the app was usable by individuals with different abilities, including those with visual or motor impairments.
    - Cross-Platform Development: Selecting a framework that supports development on both iOS and Android platforms efficiently.
    - Real-Time Performance: Implementing fast and seamless image recognition using machine learning.
    - User Engagement: Designing an interface that was visually appealing and kept users engaged through gamification.

2. The Alternatives That Were Considered
    - React Native: Considered for its strong community support but ultimately rejected due to performance limitations compared to Flutter.
    - Native Development (Swift for iOS, Kotlin for Android): Provided full platform capabilities but required separate codebases, increasing development time.
    - Custom In-House Model : Building a custom model from scratch, which would have provided more customization but increased development complexity and time.
    - Cloud-Based Solutions: Using cloud services for inference, which could offer more computational power but introduced latency , required internet connectivity and other complications.
    - AWS and Custom Server: Provided high scalability but required much more development effort and potential higher costs.
    - Firebase: Chosen for its real-time capabilities and complex integration with Flutter.

3. The Decisions That Were Made to Resolve the Issues
    - Framework Choice: Flutter was selected for its ability to create high-performance, visually appealing UIs quickly and its support for cross-platform development.
    - Machine Learning Model: A pre-trained model compatible with TensorFlow was selected to balance the ease of implementation and performance, ensuring on-device processing for fast response times.

- Backend Service: Firebase was chosen due to its comprehensive real-time database capabilities, ease of integration with Flutter, and built-in scalability. This helped streamline development and maintain user data effectively.

4. The Criteria Used to Guide Decisions

- Performance: Ensuring the app had low latency and could perform real-time image recognition smoothly.
- Development Time: Choosing tools and frameworks that allowed for rapid prototyping and shorter development cycles.
- User Experience: Ensured that design choices met accessibility standards, making the app usable for all users, including those with disabilities. This included high-contrast design and large touch targets.
- Scalability: Ensuring that the backend service could support a growing number of users without significant architectural changes.

5. The Debate Developers Went Through to Reach a Decision:

- Flutter vs. React Native: The team discussed the trade-offs between React Native's mature ecosystem and Flutter's superior performance for graphics-heavy apps. After testing both, Flutter was chosen for its smoother animations and more customizable UI components.
- Custom Model vs. Pretrained Model: Developers discussed the pros and cons of relying on cloud-based inference, which could allow for more powerful processing but would introduce network latency and require a reliable internet connection. On-device inference with TensorFlow Lite was chosen for consistent and real-time performance, ensuring that users would have the same experience regardless of connectivity.
- Cloud-Based ML vs. On-Device Inference: While cloud-based ML could potentially offer more powerful processing, the need for an internet connection and potential latency led the team to choose on-device inference for a smoother user experience.

Verification

What is Verification?

Verification refers to the process of checking if the software is meeting the specifications and needs of the clients (Altaie et al., 2020). Verification consists of two procedures: verification and validation, also referred to as V & V, where tests are conducted to ensure that the software meets requirements imposed by the client. Specifically, verification involves checking the software to ensure that it has achieved its goal in developing the product per the client's requirements without any bugs (Jiang et al., 2020). On the other hand, validation happens after the software has been checked and commonly involves a series of tests to make sure the software meets the client's requirements.

Verification Process

The verification process will be our pipeline to building robust acceptance testing for our application to ensure that each use case is developed in software to meet the requirements set by our design. This will consist of going through the steps that we believe our user will go through and ensure that at every opportunity for interaction the software acts in the expected manner. Running through our use cases step by step will ensure that we catch unexpected errors or ways in which our software does not meet requirements. For each interaction we will build edge cases that users may input into the application and ensure that no edge case leads to an unpredictable result outside of our requirements.

Furthermore, each section of the app will have a different verification process apart from End-to-end testing. End-to-end testing will be applied to the entire application by simulating real user scenarios of the workflow of the app. Apart from that, the authentication section of our application to log in and sign in users will go through Data Validation testing to ensure that the whole app validates incoming data. The machine learning model for our application will go through Model-Training-Testing which helps the model find higher accuracies and better tuning for our purposes as well as Deployment verification to ensure that our model accepts incoming post requests and answers with a predictable answer.

Test Cases

Our test cases are as follows:

1. Verify that the Firebase database supports data storage, retrieval, and maintains data integrity. To test this, we will attempt to saturate the database to experiment with its capacity and robustness, and adjust as needed.
2. Successful account creation via email, log in with 2FA, and access accounts on multiple devices. We will test edge cases for account creation with fake and real emails to verify that only real users can use the application.
3. Verify the PyTorch model is trained on 1000+ Pokémon, identifies 85% of scanned images accurately, and integrates with the camera.
4. Verify the PokeAPI fetches accurate data matching the predicted Pokémon. By utilizing unit tests we can make sure that the application replicates predictable responses from the PokeAPI.
5. Making sure the interface is responsive and adheres to web accessibility standards. This will be done by acceptance testing the features of the design of the app, in relation to accessibility standards.
6. Check that the app activates the camera and uses a bounding box to identify Pokémon through the camera in real-time. End-to-end testing can provide us with acceptance testing of the camera on our target operating systems.
7. Verify data is displayed in a consistent, accessible, and user-friendly design. We want to ensure that our app design is up to the industry standard UI/UX by following the laws of UX, such as the Aesthetic-Usability Effect, Chunking, Cognitive Bias, Cognitive Load, etc.

8. Validate that users can log in with valid credentials, access their dashboard, and receive proper error messages for incorrect details via test account.
9. Check that account deletion removes data from the database and prevents further login attempts. Through unit tests and end-to-end testing we can reach acceptance of this feature.
10. Check that admins can change usernames, track changes in logs, and notify users of updates.
11. Verify users can safely edit profile details and that changes persist across sessions on different devices (mobile, web).
12. Verify scanned Pokémon entries are saved, retrievable via Firebase, and linked to the correct user account.
13. Verify admins can ban users by username, prevent banned users from logging in, and restore access for unbanned users.
14. Verify the model assigns a score to each image, aligns the score with image quality, and displays it in the UI.
15. Verify admin and user features are clearly separated and accessible in their own permission scope.
16. Verify that predicted data sent and requested to the PokeAPI is fetched and displayed correctly.

Requirements Traceability Matrix

The following maps our test cases and status to our 16 user stories as follows:

Requirement ID	Requirement	Testcase	Development Status
1	As a Developer, I want to build/create a database with the purpose to be able to store information of users.	Verify database creation allows for data storage, successful retrieval of user data from the database, and data integrity for stored records.	Completed

2	As a user, I want to create an account using my email and 2FA so I can securely access my Pokedex.	Verify account creation via test email, successful login with 2FA, and account access from multiple devices.	Completed
3	As a developer, I want to integrate a PyTorch-based convolutional neural network model trained on 1000+ Pokémon to recognize images.	Verify the PyTorch model is trained on 900+ pokemon, the app ID's 85% of scanned images correctly, and that the model works with the model.	In-Progress
4	As a developer, I want to request data from the public PokeAPI API to serve the Pokémon data the user encounters.	Verify the PokeAPI returns requested data, and that the details match the predicted image.	In-Progress
5	As a user, I want a colorful and intuitive interface to navigate the app easily without confusion.	Verify interface responsiveness according to ADA standards.	In-Progress
6	As a user, I want to use my device’s camera to scan Pokémon images so the app can identify the Pokémon.	Verify the app can activate the camera and that the bounding box can identify Pokémon in the camera.	In-Progress
7	As a developer, I want to build a UI to display Pokémon information from the PokeAPI for user experience.	Verify data is displayed in a user-friendly layout. where the font and design is consistent all the way.	In-Progress

8	As a user, I want to access an account to view my contents and utilize the app’s features.	Verify login functionality with existing credentials, access to account dashboard, and proper error messages for incorrect credentials.	Completed
9	As a user, I want to delete an account while logged into the application, removing my information safely.	Verify account deletion removes data from the database, and that deleted accounts cannot log in.	Completed
10	As an Administrator, I want the privilege of changing a user’s username for moderation purposes.	Verify admins can change user usernames. Verify logs track username changes. Verify users are notified of the change.	Completed
11	As a user, I want to edit my account information to modify existing profile information.	Verify editing username, email, and password, and that it is safe and persists across sessions.	Completed
12	As a user, I want to save my scanned Pokémon entries to a database for easy access later.	Verify scanned Pokémon entries are stored locally, the entries are retrievable via calls to Firebase, and entries are associated with the correct account.	In-Progress
13	As an Administrator, I want the privilege of banning a user to moderate behavior.	Verify admin can ban a user by username, banned users cannot login, and unbanned users can regain account access. .	Completed

14	As a developer, I want to give the user a score for their snapped image to encourage better images.	Verify the model returns a score for each image, and that the score is displayed in the UI.	In-Progress
15	As an Administrator, I want access to an account with administrative privileges to moderate user behavior.	Verify separation of admin and user features.	Completed
16	As a developer, I want to request data from the Pokémon TCG API to display TCG cards when a Pokémon is collected.	Verify Pokémon TCG API data is fetched, collected Pokémon link to their TCG cards, and that cards display correctly in the app.	In-Progress

Reflection

Actual Timeline for Execution

The actual start of execution of the project began in mid-November, starting with initial commits on November 17 and significant work occurring between November 22 and early December. This timeframe focused on core tasks such as setting up the environment, building the model, refining the Flutter app structure, and implementing testing and validation phases. While the project's planned sprints were set to begin as early as October, much of the execution occurred closer to the end of November and into December.

Comparison with the Proposed Schedule

The proposed schedule planned for tasks to begin in early October with 4 defined sprints. Sprint 1 included database creation, account management, and the ML model which were supposed to conclude by mid-October. Sprint 2 which was scheduled for late October involved implementing the UI and integrating the database. However, most of these tasks were delayed until mid-November, resulting in their execution aligning with Sprint 3 and Sprint 4 timelines and causing all of us to adjust to working within a tighter timeframe.

Deviation from the Proposed Plan

The deviation from the proposed timeline and sprints came primarily from scheduling conflicts and time management challenges among team members. Balancing responsibilities, obligations, and varying levels of availability led to delays in starting key tasks sooner. Furthermore, dependencies between tasks—such as working camera and ML model integration—caused further delays. Despite these delays though, our team managed to meet the majority of project requirements through persistent focused collaboration and adapting the workflow to prioritize essential features.

References

1. Altaie, A. M., Al-Bayati, A. H., & Alsarraj, R. G. (2020,June.). *(PDF) verification and validation of a software: A review of the literature*. Research Gate. https://www.researchgate.net/publication/343515727_VERIFICATION_AND_VALIDATION_OF_A_SOFTWARE_A_REVIEW_OF_THE_LITERATURE

2. Atherton, G., & Cross, L. (2021, August 9). *The use of analog and digital games for autism interventions*. Frontiers in psychology. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8384560/

3. Haseler, T., & Haseler, C. (2022, February 23). *Lack of physical activity is a global problem*. BMJ (Clinical research ed.). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8864969/

4. IEEE. (2024). *Why Software Design Is Important*. IEEE Computer Society. https://www.computer.org/resources/importance-of-software-design-is-important

5. Jiang, M., et al. "An Empirical Study in Software Verification Tools." 2020 International Symposium on Theoretical Aspects of Software Engineering (TASE), 1 Dec. 2020, pp. 201–208, https://doi.org/10.1109/tase49443.2020.00035. Accessed 2 Dec. 2024.

6. Kaya, O. S., & Ercag, E. (2023, January 19). *The impact of applying challenge-based Gamification Program on students' learning outcomes: Academic achievement, motivation and flow*. Education and information technologies. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9850335/

7. Shameli, A., Althoff, T., Saberi, A., & Leskovec, J. (2017, April). *How gamification affects physical activity: Large-scale analysis of walking challenges in a mobile application*. Proceedings of the ... International World-Wide Web Conference. International WWW Conference. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5627651/