**Product Requirements**

LEGO Mindstorms Locator

| **Team** | A-Team |
| --- | --- |

# Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 04-14-2024 | 1.0 | Software Requirement | A-Team |

# Brief problem statement

The motivation for this project is accessibility. We are aiming to create an application/website to aid in finding a specific LEGO Mindstorm Block in a large pile, as well as identify a single specific piece. This would help streamline the building of Mindstorm Robots and make building projects with Mindstorm Blocks accessible to all.

The problem we are working towards fixing is the lack of availability of applications that are geared towards LEGO Mindstorm sets in finding bricks for individuals that are visually impaired, as well as easing and providing convenience for people who are interested in building these sets.

# Stakeholders

**“A-Team” University - Investor and Board of Directors**

“A-Team” university is funding this project and needs to approve the purpose and funding of this product. The investor will need to be included from day one in the product lifecycle. This can be done by scheduling a couple of presentations and meetings with the investor and board of directors which ensure they are a vital part of this project and as result, can provide us with many resources if needed.

**“A-Team” University - Product owner**

The marketing team at A-Team university is the product owner. They are responsible to have a vision of what they want to build and convey these wishes to the development team.

Below are some of the product owner responsibilities:

* Defines Features.
* Defines and accepts stories and iterations.
* Contributes to vision.
* Collaborates with Dev team.
* Prioritize needs.
* Evaluate product progress.

**Students / Teachers**

They are the end-users of the product, and they need to be included in every step to ensure project success. This will be done by surveys, interviews, and focus groups to gather their data and their feedback. Furthermore, a group of selected students and teachers will test the product before it will be launched to make sure to adjust any changes.

**Software Engineering Team**

The software engineering team is responsible for monitoring and progressing the project’s lifecycle. These are some of their responsibilities:

* Outlining the project requirements.
* Implementing the project and its features.
* Conducting software testing.
* Dividing the project into tasks and assigning these to team members.
* Closely monitoring task progression.
* Resolve any issues which delay the life cycle.

**Users**

* Must be familiar with how to manage files, like moving files from one location to another, upload and download files.
* Must be familiar with internet navigation.
* Have an updated browser.
* Must regularly check the portal for new assignments and tasks.

# System requirements

iOS (14.8+) Web/Android (8.1+) is the Operating System platform that the user must have access to in order to run the application, with access to a camera.

# Users profile

The system is to identify LEGO parts for visually impaired/blind people. The users should be able to pick up LEGOs and place them under a camera for the app to be able to identify the part.

# List of Features

F1: Identify Part Via Camera (Picture)

F2: Identify Part Via Camera (Live Feed)

F3: Identify Parts Via Camera (Picture)

F4: Identify Parts Via Camera (Live Feed)

F5: Home Screen

F6: Settings Screen

F7: Identify Part Via Color

F8: Identify Parts Via Color

F9: Upload from photo gallery

# Functional requirements (user stories)

| **No.** | **User Story Name** | **Description** | **Priority** |
| --- | --- | --- | --- |
| R1 | Identify Part Via Camera (Picture) | The user will grant the app access to the camera. The app will use the camera to scan over a pile of LEGO pieces and take a picture to identify the part that the user is looking for. | 1 |
| R2 | Identify Part Via Camera (Live Feed) | The app will ask the user what LEGO piece they are looking for. The user will grant the app access to the camera. The app will use the camera to scan over a pile of LEGO on a live feed. While the camera is in live feed mode, the app will notify the user by sound to identify the LEGO part. | 1 |
| R3 | Identify Parts Via Camera (Picture) | The user will grant the app access to the camera. The app will use the camera to scan over a pile of LEGO pieces and take a picture to identify the parts that the user is looking for. | 2 |
| R4 | Identify Parts Via Camera (Live Feed) | The app will ask the user what LEGO pieces they are looking for. The user will grant the app access to the camera. The app will use the camera to scan over a pile of LEGO on a live feed. While the camera is in live feed mode, the app will notify the user by sound to identify the LEGO parts. | 2 |
| R5 | Home Screen | The user will open the app, and the home screen will be the first screen the user will be able to see and search for various LEGO pieces. On the home screen, the user will be able to know more information about the LEGO piece, such as name, color, and quantity. In addition, the user will be able to view instructions on how to use the app. | 2 |
| R6 | Settings Screen | The user will open the app, and the home screen will be the first screen the user will see. On the bottom of the scream, the user will select the settings button to go to the settings screen. In settings, the users will be able to modify settings. | 2 |
| R7 | Identify Part(s) Via Color (Picture) | The app will ask the user what LEGO color they are looking for. The user will grant the app access to the camera. The app will use the camera to scan over a pile of LEGO and take a picture to identify the part that the user is looking for by color. | 3 |
| R8 | Identify Part(s) Via Color (Live Feed) | The app will ask the user what LEGO color they are looking for. The user will grant the app access to the camera. The app will use the camera to scan over a pile of LEGO on a live feed to identify the parts that the user is looking for by color. | 3 |

# Non-Functional Requirements

*Describe any constraints or cross-cutting characteristics of the system in a manner that is clear, specific, and testable. Each requirement should have a unique identifier (e.g., NF1, NF2...). Only present those which are applicable to your system. Categories include but are not limited to:*

| **No.** | **Non-Functional Task** | **Description** | **Priority** |
| --- | --- | --- | --- |
| NF1 | Database of Parts | The app should contain a database for all pieces needed for searching | 1 |
| NF2 | User Interface | The app should have ample screens/pages for each functional requirement | 1 |
| NF3 | Accuracy of Identifying and Locating LEGO Pieces | This app should accurately identify Lego parts 50% of the time. | 1 |
| NF4 | Responsiveness | The app should work on varying screen sizes and quick when performing actions | 2 |
| NF5 | Usability | The UI will be user-friendly and accessible to people with certain impediments. | 2 |
| NF6 | Cross-Platform Compatibility | The app should work on IOS/Android and as a web app. | 2 |
| NF7 | Accessibility | This app should allow for users to have Dark Mode, voice over, and be compatible with built-in screen readers. | 1 |

**MEASURING STRATEGIES**

NF1: Database of parts (with pictures)

* Compare the number of real parts to the number of parts stored in the database. Ensure numbers are the same.

NF2: User Interface

* Ensure there are an appropriate number of menus, pages, and options for the user to choose depending on the functionality planned.

NF3: Accuracy of identifying and locating LEGO pieces

* Record how often the app is accurate in identifying the piece that is requested (50% goal)

NF4: Responsiveness

* Ensure that the User Interface fits correctly on various screen sizes/aspect ratios.
* Measure time from user selection/press/input to completion

NF5: Usability

* Ensure the interface is easy to read and understandable.
* Measure time to find options and number of choices.

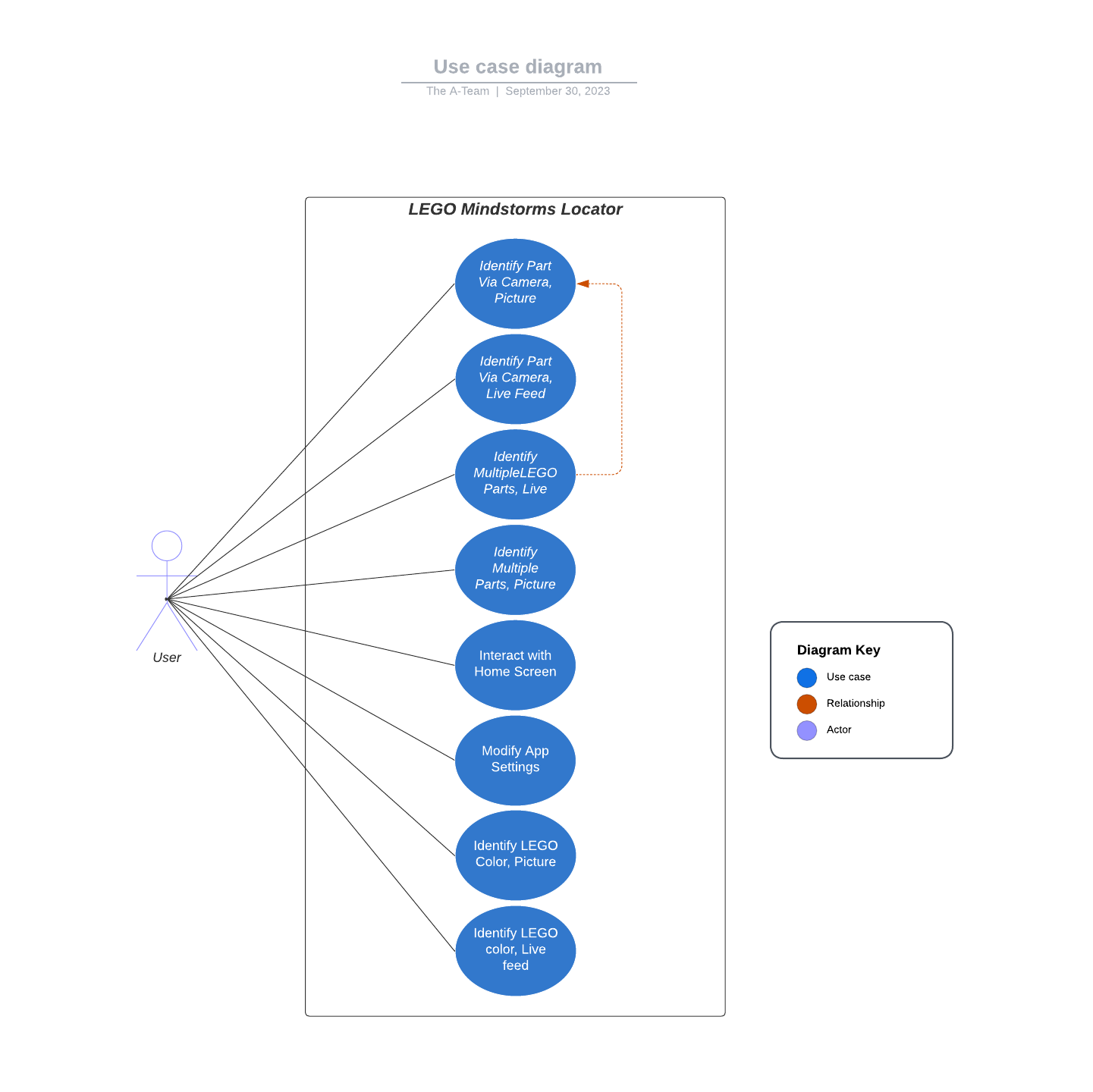
NF6: Cross-Platform Compatibility

* Ensure the app can run on various IOS/Android and that all functionality is the same across the platforms.
* Measure differences and accommodate accordingly.

NF7: Accessibility

* Measure ease of accessibility using the WCAG standards

**Use Case Diagram**



| User Case ID | R1 |
| --- | --- |
| Use Case Name | Identify Part Via Camera, Picture |
| Description | This use case allows the user to identify a specific LEGO part by capturing a picture of a pile of LEGO pieces using the app's camera functionality. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. * The user has granted camera access to the app. |
| Main Flow of Events | 1. The user selects the "Identify Part Via Camera" option. 2. The app activates the camera and captures a picture of the LEGO pile. 3. The app processes the image to identify the specific LEGO part. 4. The app displays the identification results to the user. |
| Postconditions | * The user receives information about the identified LEGO part. |
| Alternate flows | None |

**Use Case Description**

| User Case ID | R2 |
| --- | --- |
| Use Case Name | Identify Part Via Camera, Live Feed |
| Description | This use case allows the user to identify a specific LEGO part in real-time by scanning a live feed of a pile of LEGO pieces using the app's camera functionality. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. * The user has granted camera access to the app. |
| Main Flow of Events | 1. The user selects the "Identify Part Via Camera (Live Feed)" option. 2. The app activates the camera in live feed mode. 3. The app notifies the user by sound to identify the LEGO part. 4. The user scans the live feed using the camera. 5. The app processes the live feed to identify the specific LEGO part. 6. The app displays the identification results to the user. |
| Postconditions | * The user receives information about the identified LEGO part. |
| Alternate flows | None |

| User Case ID | R3 |
| --- | --- |
| Use Case Name | Identify Part Via Camera |
| Description | This use case allows the user to identify multiple LEGO parts by capturing a picture of a pile of LEGO pieces using the app's camera functionality. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. * The user has granted camera access to the app. |
| Main Flow of Events | 1. The user selects the "Identify Parts Via Camera" option. 2. The app activates the camera and captures a picture of the LEGO pile. 3. The app processes the image to identify the specific LEGO parts. 4. The app displays the identification results to the user, including information about multiple LEGO parts. |
| Postconditions | * The user receives information about the identified LEGO part. |
| Alternate flows | None |

| User Case ID | R4 |
| --- | --- |
| Use Case Name | Identify Part Via Camera |
| Description | This use case allows the user to identify multiple LEGO parts in real-time by scanning a live feed of a pile of LEGO pieces using the app's camera functionality. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. * The user has granted camera access to the app. |
| Main Flow of Events | 1. The user selects the "Identify Parts Via Camera (Live Feed)" option. 2. The app activates the camera in live feed mode. 3. The app notifies the user by sound to identify the LEGO parts. 4. The user scans the live feed using the camera. 5. The app processes the live feed to identify multiple LEGO parts. 6. The app displays the identification results to the user, including information about multiple LEGO parts. |
| Postconditions | * The user receives information about the identified LEGO part. |
| Alternate flows | None |

| User Case ID | R5 |
| --- | --- |
| Use Case Name | Home Screen |
| Description | This use case describes the user's interaction with the app's home screen, where they can search for various LEGO pieces and access information about those pieces. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. |
| Main Flow of Events | 1. The user opens the app. 2. The app displays the home screen. 3. The user can search for LEGO pieces by name, color, or other criteria. 4. The user can view information about LEGO pieces, including name, color, and quantity. 5. The user can access instructions on how to use the app. |
| Postconditions | * The user can interact with the app’s features and access information. |
| Alternate flows | None |

| User Case ID | R6 |
| --- | --- |
| Use Case Name | Settings Screen |
| Description | This use case describes the user's interaction with the app's settings screen, where they can modify app settings. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. |
| Main Flow of Events | 1. The user opens the app. 2. The app displays the home screen. 3. The user selects the settings button on the bottom of the screen. 4. The app navigates to the settings screen. 5. The user can modify app settings. |
| Postconditions | * The user can customize app settings as desired. |
| Alternate flows | None |

| User Case ID | R7 |
| --- | --- |
| Use Case Name | Identify Part(s) Via Color Picture |
| Description | This use case allows the user to identify specific LEGO parts by capturing a picture of a pile of LEGO pieces based on their color. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. * The user has granted camera access to the app. |
| Main Flow of Events | 1. The user selects the "Identify Part(s) Via Color" option. 2. The app activates the camera and captures a picture of the LEGO pile. 3. The app processes the image to identify LEGO parts based on their color. 4. The app displays the identification results to the user. |
| Postconditions | * The user receives information about the identified LEGO part based on color. |
| Alternate flows | None |

| User Case ID | R8 |
| --- | --- |
| Use Case Name | Part(s) Via Color Live Feed |
| Description | This use case allows the user to identify specific LEGO parts in real-time by scanning a live feed of a pile of LEGO pieces based on their color. |
| Actors | User |
| Preconditions | * The user has installed and launched the app. * The user has granted camera access to the app. |
| Main Flow of Events | 1. The user selects the "Identify Part(s) Via Color (Live Feed)" option. 2. The app activates the camera in live feed mode. 3. The app scans the live feed for LEGO parts based on their color. 4. The app displays the identification results to the user. |
| Postconditions | * The user receives information about the identified LEGO parts based on color. |
| Alternate flows | None |

Sponsor Requirements

I have read and approve the material in this document. If there is no external sponsor, the TA or instructor will sign it for accuracy/scope.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

Print Name Signature Date