

# Software Requirements Specification for Software Engineering: GeoWeb

Team #11, OKKM Insights

Mathew Petronilho

Oleg Glotov

Kyle McMaster

Kartik Chaudhari

April 3, 2025

# Contents

<b>1</b>	<b>Purpose of the Project</b>	<b>8</b>
1.1	User Business . . . . .	8
1.2	Goals of the Project . . . . .	8
<b>2</b>	<b>Stakeholders</b>	<b>8</b>
<b>3</b>	<b>Mandated Constraints</b>	<b>8</b>
3.1	Solution Constraints . . . . .	8
3.2	Implementation Environment of the Current System . . . . .	9
3.3	Partner or Collaborative Applications . . . . .	9
3.4	Off-the-Shelf Software . . . . .	9
3.5	Anticipated Workplace Environment . . . . .	9
3.6	Schedule Constraints . . . . .	9
3.7	Budget Constraints . . . . .	10
3.8	Enterprise Constraints . . . . .	10
<b>4</b>	<b>Naming Conventions and Terminology</b>	<b>11</b>
4.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project . . . . .	11
<b>5</b>	<b>Relevant Facts and Assumptions</b>	<b>12</b>
5.1	Relevant Facts . . . . .	12
5.2	Business Rules . . . . .	12
5.3	Assumptions . . . . .	13
<b>6</b>	<b>The Scope of the Work</b>	<b>14</b>
6.1	The Current Situation . . . . .	14
6.1.1	Data Scarcity and Quality Issues . . . . .	14
6.1.2	Industry-Specific Demands . . . . .	14
6.1.3	Technological Barriers . . . . .	14
6.1.4	Market Demand . . . . .	15
6.2	The Context of the Work . . . . .	15
6.2.1	Market Trends . . . . .	15
6.2.2	Competitive Landscape . . . . .	15
6.3	Work Partitioning . . . . .	15
6.4	Specifying a Business Use Case (BUC) . . . . .	15
6.4.1	BUC Specification: Disaster Response Data Provisioning . . . . .	16

<b>7</b>	<b>Business Data Model and Data Dictionary</b>	<b>17</b>
7.1	Business Data Model . . . . .	17
7.2	Data Dictionary . . . . .	18
<b>8</b>	<b>The Scope of the Product</b>	<b>20</b>
8.1	Product Boundary . . . . .	20
8.2	Product Use Case Table . . . . .	21
8.3	Individual Product Use Cases (PUC's) . . . . .	22
<b>9</b>	<b>Functional Requirements</b>	<b>31</b>
9.1	Functional Requirements . . . . .	31
9.2	Traceability . . . . .	37
<b>10</b>	<b>Look and Feel Requirements</b>	<b>38</b>
10.1	Appearance Requirements . . . . .	38
10.2	Style Requirements . . . . .	38
<b>11</b>	<b>Usability and Humanity Requirements</b>	<b>39</b>
11.1	Ease of Use Requirements . . . . .	39
11.2	Personalization and Internationalization Requirements . . . . .	40
11.3	Learning Requirements . . . . .	40
11.4	Understandability and Politeness Requirements . . . . .	41
11.5	Accessibility Requirements . . . . .	42
<b>12</b>	<b>Performance Requirements</b>	<b>43</b>
12.1	Speed and Latency Requirements . . . . .	43
12.2	Safety-Critical Requirements . . . . .	45
12.3	Precision or Accuracy Requirements . . . . .	45
12.4	Robustness or Fault-Tolerance Requirements . . . . .	46
12.5	Capacity Requirements . . . . .	46
12.6	Scalability or Extensibility Requirements . . . . .	47
12.7	Longevity Requirements . . . . .	47
<b>13</b>	<b>Operational and Environmental Requirements</b>	<b>48</b>
13.1	Expected Physical Environment . . . . .	48
13.2	Wider Environment Requirements . . . . .	48
13.3	Requirements for Interfacing with Adjacent Systems . . . . .	48
13.4	Productization Requirements . . . . .	50
13.5	Release Requirements . . . . .	50

<b>14 Maintainability and Support Requirements</b>	<b>52</b>
14.1 Maintenance Requirements . . . . .	52
14.2 Supportability Requirements . . . . .	52
14.3 Adaptability Requirements . . . . .	52
<b>15 Security Requirements</b>	<b>52</b>
15.1 Access Requirements . . . . .	52
15.2 Integrity Requirements . . . . .	55
15.3 Privacy Requirements . . . . .	56
15.4 Audit Requirements . . . . .	56
15.5 Immunity Requirements . . . . .	57
<b>16 Cultural Requirements</b>	<b>57</b>
16.1 Cultural Requirements . . . . .	57
<b>17 Compliance Requirements</b>	<b>57</b>
17.1 Legal Requirements . . . . .	57
17.2 Standards Compliance Requirements . . . . .	58
<b>18 Open Issues</b>	<b>59</b>
<b>19 Off-the-Shelf Solutions</b>	<b>59</b>
19.1 Ready-Made Products . . . . .	59
19.2 Reusable Components . . . . .	59
19.3 Products That Can Be Copied . . . . .	59
<b>20 New Problems</b>	<b>60</b>
20.1 Effects on the Current Environment . . . . .	60
20.1.1 Data Privacy and Security Concerns . . . . .	60
20.1.2 Environmental Footprint . . . . .	60
20.2 Effects on the Installed Systems . . . . .	60
20.2.1 Integration Challenges . . . . .	60
20.2.2 Legacy Systems Compatibility . . . . .	61
20.3 Potential User Problems . . . . .	61
20.3.1 Usability Issues . . . . .	61
20.3.2 Training and Onboarding . . . . .	61
20.3.3 Compensation and Incentives . . . . .	61
20.3.4 Technical Support . . . . .	62

20.4	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	62
20.4.1	Regulatory Constraints . . . . .	62
20.4.2	Technological Dependencies . . . . .	62
20.5	Follow-Up Problems . . . . .	62
20.5.1	Data Quality Management . . . . .	63
20.5.2	User Engagement and Retention . . . . .	63
20.5.3	Ethical Considerations . . . . .	63
20.5.4	Dependency on User Participation . . . . .	63
20.5.5	Technical Debt Accumulation . . . . .	63
<b>21</b>	<b>Tasks . . . . .</b>	<b>64</b>
21.1	Project Planning . . . . .	64
21.2	Planning of the Development Phases . . . . .	64
<b>22</b>	<b>Migration to the New Product . . . . .</b>	<b>66</b>
22.1	Requirements for Migration to the New Product . . . . .	66
22.2	Data That Has to be Modified or Translated for the New System . . . . .	66
<b>23</b>	<b>Costs . . . . .</b>	<b>66</b>
23.1	Development Costs . . . . .	66
23.1.1	Software Development . . . . .	66
23.1.2	UI/UX Design . . . . .	66
23.1.3	Testing and Quality Assurance . . . . .	67
23.2	Infrastructure Costs . . . . .	67
23.2.1	Cloud Services . . . . .	67
23.2.2	Data Acquisition . . . . .	67
23.3	Operational Costs . . . . .	67
23.3.1	User Compensation . . . . .	67
23.3.2	Maintenance and Support . . . . .	68
23.3.3	Security and Compliance . . . . .	68
23.4	Marketing and User Acquisition Costs . . . . .	68
23.4.1	Promotional Activities . . . . .	68
23.4.2	Onboarding and Training . . . . .	68
23.5	Contingency and Miscellaneous Costs . . . . .	68
23.5.1	Unexpected Expenses . . . . .	68
23.6	Budget Forecast . . . . .	68

<b>24 User Documentation and Training</b>	<b>69</b>
24.1 User Documentation Requirements . . . . .	69
24.2 Training Requirement . . . . .	70
<b>25 Undesired Event Handling</b>	<b>70</b>
<b>26 Waiting Room</b>	<b>70</b>
<b>27 Likely Changes</b>	<b>70</b>
27.1 Likely Changes . . . . .	70
<b>28 Ideas for Solution</b>	<b>71</b>

## Revision History

Date	Authors	Version
10/09/2024	Oleg, Kyle, Kartik, Mathew	Revision 0
03/08/2025	Kyle	Revision 1.0
04/01/2025	Kyle	Revision 1.1

# 1 Purpose of the Project

## 1.1 User Business

Please refer to section 1 of the Problem Statement and Goal document, found [here](#).

## 1.2 Goals of the Project

Please refer to section 4 of the Problem Statement and Goal document, found [here](#).

# 2 Stakeholders

Please refer to section 2 of the Problem Statement and Goal document, found [here](#).

# 3 Mandated Constraints

## 3.1 Solution Constraints

- **Description:** The solution must be fully compatible with the latest stable releases of Google Chrome, Firefox, Microsoft Edge, and Safari browsers.

**Rationale:** Users will interact with the web application through various modern web browsers, so ensuring cross-browser compatibility is essential for providing a consistent user experience.

**Fit Criteria:** The web application must display consistently, maintain full functionality, and support core features across all specified browsers without major visual or functional discrepancies. Testing should be conducted on each browser to validate compatibility.

**Related:** [NFR-OE6](#)



### 3.2 Implementation Environment of the Current System

There is no current environment in which our application must be implemented.

### 3.3 Partner or Collaborative Applications

There are no constraints regarding external applications that must be used alongside our product.

### 3.4 Off-the-Shelf Software

There is no required off-the-shelf software that must be used for our application.

### 3.5 Anticipated Workplace Environment

There is no particular location where users are required to work and use the product. As a web application, it can be accessed from most computers with an internet connection. We do not anticipate that the users' environment will physically constrain their ability to use the app in any way.

### 3.6 Schedule Constraints

- The proof of concept for this project must be ready to demonstrate by **November 11, 2024**. Not meeting this deadline will result in uncertainty about overcoming major risks associated with the project.
- The first project demonstration must be ready by **February 3, 2025**. Missing this deadline will reduce the time available to make refinements based on feedback and findings.
- The final demonstration must be ready by **March 24, 2025**. Missing this milestone would prevent the project from being presented and result in a significant loss of marks.

To see other documentation deadlines related to this project, refer to our [Development Plan](#).

### **3.7 Budget Constraints**

- The project budget must not exceed \$750. All funds will be sourced from the team itself.

### **3.8 Enterprise Constraints**

N/A

## 4 Naming Conventions and Terminology

### 4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Term	Definition
AC	Anticipated Change
Confidence	The measure of how likely one believes it is that a predicted value matches the true value.
Customers	Users of our application that will initiate requests for image analysis (service requests).
DAG	Directed Acyclic Graph
FR	Functional Requirement
Image Object	A subset of connected pixels in an image that has been determined by the system to be a part of a certain class, up to a level of confidence.
Label	A subset of connected pixels in an image which are all considered to be a part of the same class.
Labeller	A user of the system who is responsible to labeling objects in images for the purpose of object identification
M	Module
MG	Module Guide
NFR	Non-Functional Requirement
OS	Operating System
Req	Requirement
Service Request	Request for the system to analyze one or more related images.
Task Delegation	The assignment of tasks to workers by a central authority
UC	Unlikely Change
$\mathbb{P}(x)$	Probability of $x$

Table 1: Naming Conventions and Terminology

## 5 Relevant Facts and Assumptions

### 5.1 Relevant Facts

- **Increasing Demand for Satellite Data:** There is a growing need for accurate satellite imagery across various industries, such as agriculture, disaster response, urban planning, and environmental monitoring, driven by advancements in technology and data analysis.
- **Advances in AI and Computer Vision:** The rapid development of AI and computer vision techniques has made it possible to analyze and extract meaningful insights from large-scale satellite imagery more efficiently than before.
- **High Cost of Manual Data Labeling:** Traditional data labeling is labor-intensive, costly, and requires expert knowledge in some cases, which slows down the process of dataset creation and model training.
- **Crowdsourcing as a Viable Solution:** Crowdsourcing has proven to be a successful method for data labeling in various applications (e.g., reCAPTCHA, image tagging), providing a scalable way to collect labeled data while offering opportunities for user engagement and compensation.
- **Growing Market for Geospatial Intelligence:** Governments, private enterprises, and NGOs are increasingly investing in geospatial intelligence to support decision-making processes, making high-quality, labeled datasets more valuable for predictive modeling and operational planning.

### 5.2 Business Rules

- **Data Privacy and Confidentiality:** No confidential information is currently protected, but IP may need protection in the future as the project evolves.
- **Apache License 2.0 Usage:** The project will use the Apache License 2.0, allowing others to use and modify the software freely, with payment required if used commercially.

- **Stakeholder Communication Protocols:** There will be established communication protocols, including weekly team meetings, regular updates, and documentation of all key communications with stakeholders.
- **Workflow and Git Management:** A structured Git workflow is in place, ensuring code quality through branching strategies, approvals, and consistent naming conventions. CI/CD practices include static checks, linting, and testing before code is merged.
- **Payment and Compensation:** Compensation rules are designed to reward users for labeling data accurately and efficiently, with a clear system in place for managing payments for both users and clients.

### 5.3 Assumptions

- **Labellers Have Reliable Internet Access:** It is assumed that all users participating in labeling have stable and sufficient internet connectivity to access the platform, label data, and interact with the system without major disruptions.
- **Positive User Experience Will Attract and Retain Labellers:** The platform will rely on the assumption that a smooth user interface, fair compensation, and clear instructions will be enough to attract a substantial user base of labellers and keep them engaged over time.
- **Availability of Suitable Satellite Imagery:** It is assumed that high-resolution, relevant satellite images will be readily available from commercial or open-source providers to populate the platform and support various labeling tasks.
- **Demand for Labeled Data from Clients:** There is an expectation that industries requiring labeled satellite imagery will have a continuous and growing demand for these datasets, ensuring a steady client base for the platform's services.
- **Legal and Ethical Compliance for Data Use:** It is assumed that the acquisition and labeling of satellite imagery will not face any unforeseen legal or ethical issues, allowing for the data to be processed, shared, and monetized in accordance with existing regulations and ethical standards.

## **6 The Scope of the Work**

### **6.1 The Current Situation**

The current landscape in satellite imagery analysis is characterized by significant gaps and inefficiencies that prevent various industries from fully leveraging geospatial data.

#### **6.1.1 Data Scarcity and Quality Issues**

1. **Limited High-Quality Datasets:** There is a notable absence of comprehensive, labeled satellite imagery datasets tailored to specific industry needs. Existing datasets often lack the necessary granularity and specificity required for specialized applications.
2. **Manual Labeling Challenges:** The process of manually creating and labeling satellite imagery datasets is labor-intensive, expensive, and time-consuming. It often requires expertise that is not readily available, leading to delays and increased costs.

#### **6.1.2 Industry-Specific Demands**

1. **Diverse Use Cases:** Industries such as disaster response, environmental monitoring, urban planning, defense, agriculture, and financial analysis have unique requirements for satellite data. The lack of customized datasets limits their ability to develop accurate and effective computer vision models.
2. **Operational Inefficiencies:** Without tailored datasets, organizations face difficulties in automating image analysis tasks, resulting in slower decision-making processes and reduced operational effectiveness.

#### **6.1.3 Technological Barriers**

1. **Complexity of Data Processing:** Processing raw satellite images to extract meaningful insights involves complex algorithms and significant computational resources, which many organizations are unable to access.

#### **6.1.4 Market Demand**

1. Growing Need for Geospatial Intelligence: As industries increasingly recognize the value of geospatial data, the demand for high-quality, actionable satellite imagery insights is rising. However, the supply side has not kept pace with this growing demand.

### **6.2 The Context of the Work**

The development of the OKKM Insights platform occurs within a broader context of technological advancements, market trends, and evolving industry needs.

#### **6.2.1 Market Trends**

1. Demand for Real-Time Insights: There is a growing expectation for real-time or near-real-time insights derived from satellite imagery to support timely decision-making in critical areas such as disaster response and environmental monitoring.

#### **6.2.2 Competitive Landscape**

1. Opportunities for Innovation: The intersection of AI, crowdsourcing, and satellite imagery presents opportunities for innovative solutions that can address existing gaps in the market, particularly in niche and high-demand use cases.

### **6.3 Work Partitioning**

The division of labour is thoroughly described in the Development plan document.

### **6.4 Specifying a Business Use Case (BUC)**

To demonstrate the practical application and value proposition of the OKKM Insights platform, it is essential to define specific Business Use Cases (BUCs) that illustrate how different stakeholders will interact with and benefit from the platform.

#### **6.4.1 BUC Specification: Disaster Response Data Provisioning**

A government search and rescue agency receives a distress call reporting a downed airplane in a remote or inaccessible area. This report necessitates immediate action to locate the wreckage, assess the situation, and coordinate rescue operations. The OKKM Insights platform is notified of the incident through an integrated communication channel linked to the SAR agency’s reporting system. The platform automatically acquires the latest satellite images of the reported area from third-party providers and segments the large images into smaller, manageable tiles focused on the incident area. The algorithm then preprocesses and creates a task for data labellers using intelligent task delegation algorithms. Data labellers annotate critical features such as the airplane wreckage, debris fields, potential survivors, and any signs of environmental hazards. Consensus algorithms cross-verify annotations from multiple labellers, ensuring high accuracy and consistency. The platform then generates actionable insights, such as the precise location of the wreckage, extent of debris spread, accessibility of the area, and potential hazards. This data can then be used to train computer vision models using the newly labeled data to enhance the platform’s capability to automatically detect and classify aircraft and related features in satellite images. Finally the platform provides the SAR agency with access to the labeled dataset and the trained models through the platform’s user interface.



## 7 Business Data Model and Data Dictionary

### 7.1 Business Data Model



Figure 1: Business Data Model

## 7.2 Data Dictionary

Data Object	Content	Type
Labeller	LabellerID	Class
Customer	CustomerID	Class
Labeller Payment Details	LabellerID Payment Details	Class
Customer Payment Details	CustomerID Payment Details	Class
Labeller Personal Details	LabellerID Personal Details	Class
Customer Personal Details	CustomerID Personal Details	Class
Historic Accuracy	LabellerID Object Class Accuracy Samples	Class
Service Request	Image CustomerID Request Deadline	Class
Image	Image Data Image Metadata	Class
Label	Labeled Class Bounding Box Labeller Label Metadata	Class
Object Class	Object Class List Object Class Score Labels Image	Class

Task Allocation	{Service Request +Available Labellers +Historic Accuracy +Object Class +Request Deadline }	Data Flow
Class Consensus	{Label +Historic Accuracy}	Data Flow
Payment Details	All details necessary to process payment. Depends on payment vendor and location	Attribute/Element
Personal Details	Details necessary to identify and contact a user	Attribute/Element
Accuracy	Represents proportion of correct labels $a \in \mathbb{R} \wedge 0 \leq a \leq 1$	Attribute/Element
Samples	$n \in \mathbb{N} \wedge a \geq 0$	Attribute/Element
Request Deadline	YYYY-MM-DDTHH:MM:SS In 24HR UTC	Attribute/Element
Image Data	NxM grid of RGB pixels	Attribute/Element
Image Metadata	Additional details related to an image	Attribute/Element
Label class	String representing class of object	Attribute/Element
Bounding Box	$(x_0, y_0), (x_1, y_1)$ where $\forall x_j, y_j : 0 \leq x_j \leq N \wedge 0 \leq y_j \leq M$	Attribute/Element
Label Metadata	Additional details related to an label	Attribute/Element
Object Class List	List of classes reported in image	Attribute/Element
Object Class Score	Likelihood of class appearing in image $s \in \mathbb{R} \wedge 0 \leq s \leq 1$	Attribute/Element

## 8 The Scope of the Product

### 8.1 Product Boundary

The use case diagram depicted below identifies the boundaries between the users and the product.



Figure 2: Use Case diagram

## 8.2 Product Use Case Table

PUC No	PUC Name	Actor/s	Input & Output
1	Sign Up	Labellers, Customers	Personal information & credentials (in), New account (out)
2	Log In	Labellers	Credentials (in)
3	Customer Log In	Customers	Credentials (in)
4	Update User Information	Labellers, Customers	Information to change (in), Updated information (out)
5	Annotate Images	Labellers	Image & annotation (in), Labeled image (out)
6	Segment Annotation	Labellers	Segmented image & user selections (in), Image with labeled segments (out)
7	Bounding Box Annotation	Labellers	Image & box annotations (in), Image with labeled boxes (out)
8	Classification	Labellers	Image & choices (in), Image labeled with the selected choice (out)
9	Cash Out	Labellers, Financial Institution	Financial information (in), Payment to user (out)
10	View Image Insights	Customers	Request to view insights (in), Image insights (out)
11	Export Results	Customers	Request to export results (in), Results in downloadable format (out)
12	Request Image Analysis	Customers	Labels, images & examples (in)

13	Upload Images	Customers	Images (in)
14	Request Images	Customers, Satellite Image Provider	Description of images wanted (in), Images (out)
15	Process Payment	Customers, Financial Institution	Financial information (in), Payment to app (out)

Table 2: Product Use Cases

### 8.3 Individual Product Use Cases (PUC's)

#### PUC0 - Sign Up

**Primary Actors:** Labellers, Customers

**Precondition:** None

**Trigger:** User selects create an account

**Success Scenario:**

1. User provides the required information and agrees with the privacy policy of the application
2. System verifies all required information has been provided and is of valid syntax
3. System securely registers the given information
4. User is directed to the login page

**Post-condition:** User has successfully created an account and the application has securely stored their information

**Related:** [FR0](#)

#### PUC1 - Log In

**Primary Actors:** Labellers

**Precondition:** Labeller has signed up for an account

**Trigger:** Labeller selects log in

**Success Scenario:**

1. Labeller enters credentials
2. System verifies all required information has been provided and matches the stored records
3. Labeller is redirected to the home page as a logged-in user

**Post-condition:** The labeller has access to their account. All account related information, such as labeling progress, is accurately represented

**Related:** [FR10](#)

### **PUC2 - Customer Log In**

**Primary Actors:** Customers

**Precondition:** Customer has signed up for an account

**Trigger:** Customer selects log in

**Success Scenario:**

1. Customer enters credentials
2. System verifies all required information has been provided and matches the stored records
3. Customer is redirected to the home page as a logged-in customer, which gives them access to more features than a labeller

**Post-condition:** The customer has access to their account and all their current image analysis projects

**Related:** [FR1](#)

### **PUC3 - Update User Information**

**Primary Actors:** Labellers, Customers

**Precondition:** User has logged in

**Trigger:** User selects update information

**Success Scenario:**

1. User enters updated information
2. System verifies the new information has valid syntax
3. System updates the information

4. User is given a confirmation that their information has been updated

**Post-condition:** The information that the system stores about the user has been updated

**Related:** [FR2](#), [FR11](#)

#### **PUC4 - Annotate Satellite Images**

**Primary Actors:** Labellers

**Precondition:** Labeller has logged in

**Trigger:** Labeller selects a listed project

**Success Scenario:**

1. A satellite image related to the selected project and a description of what to annotate is shown to the labeller
2. The labeller annotates the satellite image to the best of their ability
3. The annotated satellite image is submitted by the labeller
4. The labeller is rewarded and the satellite image is stored by the system
5. If the labeller skips the image rather than submits it, then the annotated image will not be stored by the system and the labeller will not receive a reward
6. The next satellite image of the selected project is shown to the labeller and the process repeats

**Post-condition:** All submitted annotated satellite images are stored by the system. The labeller's reward balance is updated to reflect their submissions

**Related:** [FR13](#)

#### **PUC5 - Segment Annotation**

**Primary Actors:** Labellers

**Precondition:** Labeller has logged in

**Trigger:** Labeller selects a listed project

**Success Scenario:**



1. The same as Annotate Images, but the labeller is given a satellite image segmented into different parts and must label each segment to the best of their ability

**Post-condition:** All submitted annotated satellite images are stored by the system. The labeller's reward balance is updated to reflect their submissions

**Related:** [FR13](#)

### **PUC6 - Bounding Box Annotation**

**Primary Actors:** Labellers

**Precondition:** Labeller has logged in

**Trigger:** Labeller selects a listed project

**Success Scenario:**

1. The same as Annotate Images, but the labeller must annotate areas of the satellite image by drawing boxes and providing a label for each box

**Post-condition:** All submitted annotated satellite images are stored by the system. The labeller's reward balance is updated to reflect their submissions

**Related:** [FR13](#)

### **PUC7 - Classification**

**Primary Actors:** Labellers

**Precondition:** Labeller has logged in

**Trigger:** Labeller selects a listed project

**Success Scenario:**

1. The same as Annotate Images, but the labeller must classify what category a satellite image belongs to given several options

**Post-condition:** All submitted annotated satellite images are stored by the system. The labeller's reward balance is updated to reflect their submissions

**Related:** [FR13](#)

### **PUC8 - Cash Out**

**Primary Actors:** Labellers

**Secondary Actor:** Financial Institution

**Precondition:** Labeller has logged in and their reward balance is at least \$1

**Trigger:** Labeller selects cash out

**Success Scenario:**

1. The system retrieves the labeller's payment information
2. The labeller confirms it is correct, or updates their information accordingly
3. The system sends a payment to the labeller through their financial institution
4. The labeller's reward balance is set back to \$0

**Post-condition:** The financial account that the labeller provided has gained money equal to the labeller's reward balance

**Related:** [FR12](#)

## **PUC9 - View Image Insights**

**Primary Actors:** Customers

**Precondition:** Customer has logged in and has at least one satellite image analysis project

**Trigger:** Customer selects insights for a project

**Success Scenario:**

1. The system provides the customer with a progress update, key statistics, and an overview of current labels for specific satellite images
2. The customer selects a specific image to view insights
3. The system gets the information it has stored related to the image and provides insights such as confidence level of annotations

**Post-condition:** The customer has knowledge about how their project is progressing and if the labeling of the satellite images has been complete

**Related:** [FR5](#), [FR14](#)

## **PUC10 - Export Results**

**Primary Actors:** Customers

**Precondition:** Customer has logged in, has at least one satellite image analysis project, and all labeled satellite images in that project have an acceptable accuracy

**Trigger:** Customer selects export data

**Success Scenario:**

1. The system retrieves all the labeled satellite images associated with the project
2. The system asks the customer where they would like the images to be saved
3. The customer specifies the save location
4. The images are downloaded to the specified location

**Post-condition:** The customer has complete access to the labeled dataset of satellite images

**Related:** [FR5](#), [FR14](#)

## **PUC11 - Request Image Analysis**

**Primary Actors:** Customers

**Precondition:** Customer has logged in

**Trigger:** Customer creates a new project request

**Success Scenario:**

1. The system prompts the customer with a form to fill out regarding information about the project such as what is to be labeled
2. The customer fills out the form and submits it
3. The system confirms the forms information is valid. It prompts the customer to upload the satellite image dataset to be analyzed
4. The customer uploads their own satellite images or requests the system to get images for them

5. The system asks the customer to provide sample labeled images that can be shared with labellers
6. The customer uploads sample labeled images
7. Based on the size and difficulty of the project, the system provides the total cost of the project to the customer
8. The customer provides their payment details
9. The system processes the payment
10. The system makes the project available to labellers so they can annotate the satellite images

**Post-condition:** The image analysis project has been created and labellers can start annotating the satellite images

**Related:** [FR4](#)

## **PUC12 - Upload Satellite Images**

**Primary Actors:** Customers

**Precondition:** Customer has logged in and has created a new project request

**Trigger:** Customer submits the information form for a new project

**Success Scenario:**

1. The system prompts the customer to upload the images they would like to be analyzed
2. The customer selects the images they want analyzed and uploads them
3. The system stores the images with a reference to the project

**Post-condition:** The system has all the images that will be analyzed for the project

**Related:** [FR6](#)

### **PUC13 - Request Satellite Images**

**Primary Actors:** Customers

**Secondary Actors:** Satellite Image Provider

**Precondition:** Customer has logged in and has created a new project request

**Trigger:** Customer submits the information form for a new project

**Success Scenario:**

1. The system prompts the customer about the type of satellite images they want. This could be satellite images from a specific time frame in a specific area
2. The customer specifies the type of satellite images they want for analysis
3. The system contacts a satellite image provider, and pays a fee to them in exchange for the images. The fee is charged to the customer
4. The system stores the images with a reference to the project

**Post-condition:** The system has all the satellite images that will be analyzed for the project

**Related:** [FR7](#)

### **PUC14 - Process Payment**

**Primary Actors:** Customers

**Secondary Actor:** Financial Institution

**Precondition:** Customer has logged in and has created a new project request

**Trigger:** Customer has filled out all the details of the project and uploaded all the photos

**Success Scenario:**

1. The system retrieves the customer's payment information
2. The customer confirms it is correct, or updates their information accordingly
3. The system sends a payment request to the customer's financial institution

4. The financial institution validates the payment information and sends the payment
5. The payment is deposited in the financial account associated with the system

**Post-condition:** The fees for facilitating the project have been paid

**Related:** [FR3](#)

## Context Diagram



Figure 3: Context Diagram

## 9 Functional Requirements

### 9.1 Functional Requirements

#### FR0

- **Priority:** High
- **Description:** The system shall allow Customers to create a new user account.
- **Rationale:** The customers must have an account to request the services provided by the system.
- **Fit Criterion:** Customer is created in and stored in the database.
- **Related:** [PUC0](#), [NFR-PR0](#), [NFR-SE2](#)

#### FR1

- **Priority:** High
- **Description:** The system shall allow customers with an existing account to authenticate themselves using the credentials stored by the system.
- **Rationale:** To access privileged information, the system must authenticate customers.
- **Fit Criterion:** Customer can use correct credentials to log in to user account.
- **Related:** [PUC2](#), [NFR-SE1](#)

#### FR2

- **Priority:** Medium
- **Description:** The system shall allow authenticated Customers to edit their account information.
- **Rationale:** After creating an account, the customers must have the necessary tools to modify the information they have provided.

- **Fit Criterion:** After logging in, customers can modify the stored personal information.
- **Related:** [PUC3](#), [NFR-UH0](#)

### FR3

- **Priority:** Out of Scope
- **Description:** The system shall accept payments from authenticated Customers.
- **Rationale:** To request services, the system shall integrate with payment services.
- **Fit Criterion:** After logging in, customers can send a payment to the system.
- **Related:** [PUC14](#), [NFR-OE2](#), [NFR-SE9](#)

### FR4

- **Priority:** High
- **Description:** The system shall accept service requests from authenticated Customers.
- **Rationale:** The system's key offering is the ability to request services.
- **Fit Criterion:** After logging in, customers can send a service request which is accepted by the system.
- **Related:** [PUC11](#), [NFR-PR7](#), [NFR-SE1](#)

### FR5

- **Priority:** High
- **Description:** The system shall provide reports to authenticated customers when a service request is complete.
- **Rationale:** The system must provide the results of a service request to the users who initiated it.



- **Fit Criterion:** After logging in and after the completion of a service request, customers can receive a service request report.
- **Related:** [PUC9](#), [PUC10](#), [NFR-PR2](#)

## FR6

- **Priority:** High
- **Description:** The system shall allow an authenticated Customer to upload images associated with a service request.
- **Rationale:** If the customer has images to be analyzed, they must have the ability to upload them to the system.
- **Fit Criterion:** After logging in and after the initiation of a service request, customers can add images associated to the service request.
- **Related:** [PUC12](#), [NFR-PR8](#)

## FR7

- **Priority:** Medium
- **Description:** The system shall allow a customer to request satellite images for a given geographic area, as designated using a specified geographic coordinate system.
- **Rationale:** If customers do not have images for the area they would like to be analyzed, they can request the system to source them on their behalf. The system may not be able to fulfill the request.
- **Fit Criterion:** After logging in and after the initiation of a service request, customers can request the system to locate images from external providers by providing geographic coordinates.
- **Related:** [PUC13](#), [NFR-PR8](#), [NFR-OE1](#)

## FR8

- **Priority:** Medium
- **Description:** The system shall alert a customer if their service request is not able to be fulfilled.
- **Rationale:** In the event the system is unable to complete a request, it should alert the customer, so they are aware.
- **Fit Criterion:** After logging in and after the initiation of a service request, the customer will get an alert if the request is unable to be fulfilled.
- **Related:** [NFR-SE4](#)

## FR9

- **Priority:** High
- **Description:** The system shall allow Labellers to create a new user account.
- **Rationale:** The Labellers must have an account to request the services provided by the system.
- **Fit Criterion:** Labeller is created in and stored in the database.
- **Related:** [PUC0](#), [NFR-PR1](#), [NFR-SE2](#), [NFR-SE3](#)

## FR10

- **Priority:** High
- **Description:** The system shall allow Labellers with an existing account to authenticate themselves using the credentials stored by the system.
- **Rationale:** In order to access privileged information, the system must Labellers customers.
- **Fit Criterion:** Labellers can use correct credentials to log in to user account.

- **Related:** [PUC1](#), [NFR-SE0](#)

#### FR11

- **Priority:** Medium
- **Description:** The system shall allow authenticated Labellers to edit their account information.
- **Rationale:** After creating an account, the Labellers must have the necessary tools to modify the information they have provided.
- **Fit Criterion:** After logging in, Labellers can modify the stored personal information.
- **Related:** [PUC4](#), [NFR-UH0](#)

#### FR12

- **Priority:** Out of Scope
- **Description:** The system shall allow Labellers to request their accrued earnings be transferred to their provided banking platform.
- **Rationale:** The system must ensure Labellers are compensated for their efforts.
- **Fit Criterion:** After logging in, Labellers can request their earnings be deposited in the account stored in their personal information.
- **Related:** [PUC8](#), [NFR-PR4](#), [NFR-OE2](#), [NFR-SE9](#)

#### FR13

- **Priority:** High
- **Description:** The system shall allow Labellers to annotate images displayed to them with the purpose of collecting annotation data.
- **Rationale:** This is the key service the system provides.
- **Fit Criterion:** Labellers will be able to annotate images.

- **Related:** [PUC4](#), [PUC5](#), [PUC6](#), [PUC7](#), [NFR-UH2](#), [NFR-UH9](#) [NFR-PR3](#), [NFR-SE0](#)

#### **FR14**

- **Priority:** **High**
- **Description:** The system shall combine annotations obtained from authenticated Labellers into a consolidated report.
- **Rationale:** This is the key service the system provides.
- **Fit Criterion:** After a service request has been initiated, and the system has determined the label accuracy is over a certain threshold.
- **Related:** [PUC9](#), [PUC10](#) [NFR-PR5](#)

## 9.2 Traceability

PUC #	Functional Requirement														
	FR0	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10	FR11	FR12	FR13	FR14
PUC0	X									X					
PUC1											X				
PUC2		X													
PUC3			X									X			
PUC4														X	
PUC5														X	
PUC6														X	
PUC7														X	
PUC8													X		
PUC9						X									X
PUC10						X									X
PUC11					X										
PUC12							X								
PUC13								X							
PUC14				X											

Table 3: Traceability between FR and PUC

NFR #	Functional Requirement														
	FR0	FR1	FR2	FR3	FR4	FR5	FR6	FR7	FR8	FR9	FR10	FR11	FR12	FR13	FR14
UH0			X									X			
UH2														X	
UH9														X	
PR0	X														
PR1										X					
PR2						X									
PR3														X	
PR4													X		
PR5															X
PR7					X										
PR8							X	X							
OE1								X							
OE2				X									X		
SE0											X			X	
SE1		X			X										
SE2	X									X					
SE3	X									X					
SE4									X						
SE9				X									X		

Table 4: Traceability between FR and NFR

## 10 Look and Feel Requirements

### 10.1 Appearance Requirements

#### NFR-LF0

- **Priority:** Medium
- **Description:** The application shall adapt to various screen sizes, ensuring legibility and an uncluttered layout.
- **Rationale:** Users will have computers with varying screen sizes, so a consistent experience across all these sizes is ideal.
- **Fit Criterion:** Visual elements must not exceed the boundaries of a screen with a size between the range 1024×768 pixels to 1920×1080 pixels.

#### NFR-LF1

- **Priority:** Medium
- **Description:** Interactive elements such as buttons shall provide visual feedback to the user.
- **Rationale:** This will allow users a better understanding of when their actions have been processed by the application.
- **Fit Criterion:** Every interactive element changes colour or displays additional visual cues, such as animations or shadows, to indicate interaction.

### 10.2 Style Requirements

#### NFR-LF2

- **Priority:** Medium
- **Description:** The application should maintain a unified visual design across all components.

- **Rationale:** A consistent appearance enhances the application's cohesiveness and conveys a professional aesthetic.
- **Fit Criterion:** Font type, sizing, and colour, along with background tones are all consistent throughout the application.

## 11 Usability and Humanity Requirements

### 11.1 Ease of Use Requirements

#### NFR-UH0

- **Priority:** High
- **Description:** The platform will have a well-organized navigation menu, allowing users to quickly find labeling tasks, settings, help, and account details.
- **Rationale:** An easy-to-navigate platform helps reduce the time taken for users to find what they need, improving the efficiency of labeling tasks and overall user satisfaction.
- **Fit Criterion:** At least 90% of users should be able to navigate to any main feature within 3 clicks, as determined by usability testing.
- **Related:** [FR2](#), [FR11](#)

#### NFR-UH1

- **Priority:** Medium
- **Description:** The platform's buttons, forms, and other UI elements should follow a consistent style and behavior throughout, to ensure a predictable user experience.
- **Rationale:** Consistency reduces cognitive load and helps users feel comfortable and confident when interacting with the system, as they can easily recognize and understand patterns in the UI.
- **Fit Criterion:** Consistency will be validated by usability tests where at least 95% of users recognize and correctly use recurring interface elements across different pages.

## 11.2 Personalization and Internationalization Requirements

### NFR-UH2

- **Priority:** Low
- **Description:** Users can set preferences to prioritize specific types of labeling tasks (e.g., agricultural, urban) and receive suggestions aligned with their interests and expertise.
- **Rationale:** Allowing users to work on tasks of interest or within their expertise can improve task accuracy and user satisfaction, leading to better retention and data quality.
- **Fit Criterion:** Users should be able to set their task preferences in their profile, with at least 80% of their assigned tasks matching these preferences over time.
- **Related:** [FR13](#)

### NFR-UH3

- **Priority:** Out of Scope
- **Description:** The platform should display localized formats for date, time, and currency based on the user's location or preference settings.
- **Rationale:** Displaying culturally familiar formats enhances user comfort and reduces errors when interpreting critical information like deadlines or earnings.
- **Fit Criterion:** Date, time, and currency formats should adjust automatically based on the user's location or manual settings, verified through localized testing.

## 11.3 Learning Requirements

### NFR-UH4

- **Priority:** High



- **Description:** Users can complete a simple tutorial to learn the platform features.
- **Rationale:** A tutorial can improve understanding and decrease frustration in users.
- **Fit Criterion:** At least 80% of users should find it helpful, based on post-training feedback surveys.

#### NFR-UH5

- **Priority:** Medium
- **Description:** Provide users with simulated labeling tasks to practice without affecting actual datasets, allowing them to learn through hands-on experience before starting real work.
- **Rationale:** Practice tasks help users understand the labeling process without pressure, improving their confidence and accuracy before handling real data.
- **Fit Criterion:** At least 85% of users should complete at least one practice task before labeling real data, with a satisfaction rate of 90% based on feedback.

## 11.4 Understandability and Politeness Requirements

#### NFR-UH6

- **Priority:** Out of Scope
- **Description:** Contextual help pop-ups should be available throughout the platform, offering brief explanations for various features and guidance on how to complete tasks.
- **Rationale:** Providing help exactly where users need it reduces confusion and makes it easier for users to understand how to use different parts of the platform without extensive searching.
- **Fit Criterion:** At least 90% of users should find the contextual help pop-ups clear and helpful during usability testing.

## NFR-UH7

- **Priority: Medium**
- **Description:** Error messages will not only indicate what went wrong but also provide actionable steps to resolve the issue in a friendly, non-blaming tone.
- **Rationale:** Friendly and constructive error handling reduces user frustration and helps users quickly resolve issues without needing support, improving overall experience.
- **Fit Criterion:** At least 95% of errors will be accompanied by actionable instructions, and users should report a low frustration rate (< 10%) when encountering errors, based on feedback.

## 11.5 Accessibility Requirements

### NFR-UH8

- **Priority: Out of Scope**
- **Description:** The platform will provide options for users to adjust text size and choose between different color themes (e.g., light, dark, high contrast) to improve readability based on their preferences.
- **Rationale:** Allowing adjustments for visual elements helps users with low vision, color blindness, or different environmental lighting conditions to comfortably use the platform.
- **Fit Criterion:** At least three different color themes will be available, and users should be able to increase text size up to 200% without loss of content or functionality.

### NFR-UH9

- **Priority: Medium**
- **Description:** The platform will support keyboard shortcuts for essential actions (e.g., navigating between tasks, submitting labels) to facilitate quick access and improve usability for users who cannot use a mouse.

- **Rationale:** Keyboard shortcuts provide an alternative way to interact with the system, improving efficiency for power users and ensuring accessibility for users with limited mobility.
- **Fit Criterion:** All core actions should have keyboard shortcuts available, and usability tests should confirm that at least 95% of actions are accessible via keyboard navigation.
- **Related:** [FR13](#)

## 12 Performance Requirements

### 12.1 Speed and Latency Requirements

#### NFR-PR0

- **Priority:** Medium
- **Description:** The system shall process new user requests within 15 minutes of a Customer completing the account creation process 90% of the time, and within 48 hours in all cases.
- **Rationale:** The user likely has urgent needs if they are signing up to access the system. They must have quick access to the services offered by the system, and authentication is the first step of this process.
- **Fit Criterion:**  
Let  $t_{\text{newCustomerAccount}}$  be the time it takes to process a new Customer account request, in hours.  
 $\mathbb{P}(t_{\text{newCustomerAccount}} < 0.25) \geq .90 \wedge \mathbb{P}(t_{\text{newCustomerAccount}} < 48) = 1$
- **Related:** [FR0](#)

#### NFR-PR1

- **Priority:** Medium
- **Description:** The system shall process new user requests within 15 minutes of a Labeller completing the account creation process 90% of the time, and within 48 hours in all cases.

- **Rationale:** To improve engagement from Labellers, there should be minimal delay in getting started.
- **Fit Criterion:**  
Let  $t_{\text{newLabelAccount}}$  be the time it takes to process a new Labeller account request, in hours.  
 $\mathbb{P}(t_{\text{newLabelAccount}} < 0.25) \geq 0.9 \wedge \mathbb{P}(t_{\text{newLabelAccount}} < 48) = 1$
- **Related:** [FR9](#)

## NFR-PR2

- **Priority:** High
- **Description:** The system shall return a complete report of results to the Customer within the negotiated amount of time, 90% of the time, and within 48 additional hours in all cases.
- **Rationale:** The user is expecting to be able to act on the insights the system provides. To build trust and loyalty, the system must ensure that it is meeting the agreed upon timelines.
- **Fit Criterion:**  
Let  $t_{\text{serviceRequestTime}}$  be the time it takes to complete a service request, in hours.  
Let  $t_{\text{serviceRequestTimeLimit}}$  be the negotiated time limit for completing a service request, in hours.  
 $\mathbb{P}(t_{\text{serviceRequestTime}} < t_{\text{serviceRequestTimeLimit}}) \geq 0.9$   
 $\wedge \mathbb{P}(t_{\text{serviceRequestTime}} < t_{\text{serviceRequestTimeLimit}} + 48) = 1$
- **Related:** [FR5](#), [NFR-PR7](#)

## NFR-PR3

- **Priority:** High
- **Description:** The system shall take no longer than 10 seconds to display the next image to be labeled to a labeller, if there is one available.

- **Rationale:** To reduce friction for labellers, the system must ensure there is no unnecessary delay in preparing the next job. For users exposed to modern media, attention spans can be expected to be less than 10 seconds [7].
- **Fit Criterion:**  
Let  $t_{\text{imageServing}}$  be the time it takes to serve the next image, in seconds.  
Let  $x_{\text{nextImage}}$  equal True if there is a next image available, and False otherwise.  
 $x_{\text{nextImage}} \Rightarrow t_{\text{imageServing}} \leq 10$
- **Related:** [FR13](#)

#### NFR-PR4

- **Priority:** Out of Scope
- **Description:** The system shall deliver earned payouts to labellers within 7 business days of a request being made through the system.
- **Rationale:** The labellers are entitled to their earned income, and there must not be unnecessary delay. 7 days accounts for delays in the platform used to distribute payments.
- **Fit Criterion:**  
Let  $t_{\text{payoutDelay}}$  be the time it takes for a user to receive their payments after a request is received by the system, in days.  
 $t_{\text{payoutDelay}} < 7$
- **Related:** [FR12](#)

## 12.2 Safety-Critical Requirements

N/A

## 12.3 Precision or Accuracy Requirements

#### NFR-PR5

- **Priority:** High

- **Description:** The system shall report accurate labels 75% of the time.
- **Rationale:** Average data label accuracy for competitors is greater than 75% [8]. The system should at a minimum provide the same label accuracy.
- **Fit Criterion:**  
 Let  $O$  be the set of objects to label.  
 Let  $C$  be the set of classes an object in  $O$  can be.  
 Let  $L_{\text{True}} : O \rightarrow C$  be a function which maps objects in  $O$  to their true classes in  $C$ . Let  $L_{\text{Guess}} : O \rightarrow C$  be the function derived from the system which maps objects in  $O$  to their assumed classes in  $C$ .  
 $(\forall o \in O \mid \mathbb{P}(L_{\text{True}}(o) = L_{\text{Guess}}(o)) \geq 0.75)$
- **Related:** [FR14](#)

## 12.4 Robustness or Fault-Tolerance Requirements

### NFR-PR6

- **Priority:** Low
- **Description:** The system shall have 97% uptime.
- **Rationale:** It is crucial that in emergency response use cases, the system is able to accept and process requests with minimal delay.
- **Fit Criterion:**  
 Let  $t_{\text{uptime}}$  be the uptime of the system.  
 Let  $t_{\text{downtime}}$  be the downtime of the system.  
 $\frac{t_{\text{uptime}}}{t_{\text{uptime}} + t_{\text{downtime}}} > 0.97$

## 12.5 Capacity Requirements

### NFR-PR7

- **Priority:** Low
- **Description:** The system shall have the capacity to support enough labellers to meet all service request deadlines.

- **Rationale:** This requirement is critical to satisfy NFR-PR2.
- **Fit Criterion:** See [NFR-PR2](#).
- **Related:** [FR4](#), [NFR-PR2](#), [NFR-PR9](#)

#### NFR-PR8

- **Priority:** Medium
- **Description:** The system shall have the capacity to store and process large image files.
- **Rationale:** This requirement is necessary to obtain information from satellite images, which can be several gigabites in size.
- **Fit Criterion:** The system will not crash or fail to store when given images files <50 Gb in size.
- **Related:** [FR6](#), [FR7](#)

## 12.6 Scalability or Extensibility Requirements

#### NFR-PR9

- **Priority:** Low
- **Description:** The system shall be able to scale to meet the capacity specified in NFR-PR7.
- **Rationale:** This requirement is critical to satisfy NFR-PR7.
- **Fit Criterion:** See [NFR-PR7](#).
- **Related:** [NFR-PR7](#)

## 12.7 Longevity Requirements

N/A

## 13 Operational and Environmental Requirements

### 13.1 Expected Physical Environment

N/A

### 13.2 Wider Environment Requirements

NFR-OE0

- **Priority:** Out of Scope
- **Description:** The system shall implement energy-efficient practices in cloud usage and server management.
- **Rationale:** Minimizes the platform's carbon footprint and promotes sustainability.
- **Fit Criterion:** Cloud infrastructure usage is optimized to reduce energy consumption by at least 20% compared to baseline measurements.

### 13.3 Requirements for Interfacing with Adjacent Systems

NFR-OE1

- **Priority:** Low
- **Description:** The system shall support standardized APIs and data formats for the automatic acquisition and integration of satellite images from third-party providers.
- **Rationale:** Facilitates seamless integration and efficient data handling from satellite data providers.
- **Fit Criterion:** The platform can successfully ingest data from at least two major satellite data providers using their APIs without manual intervention.
- **Related:** [FR7](#)



## NFR-OE2

- **Priority:** Out of Scope
- **Description:** The system shall integrate with reliable and secure payment processors to handle user compensations and client payments efficiently.
- **Rationale:** Ensures secure and efficient financial transactions, building trust with users and clients.
- **Fit Criterion:** Transactions are processed securely through integrated payment gateways, achieving a transaction success rate of 99.5%.
- **Related:** [FR3](#), [FR12](#)

## NFR-OE3

- **Priority:** Out of Scope
- **Description:** The system shall support multiple currencies to accommodate a global user base and international clients.
- **Rationale:** Enhances the platform's accessibility and usability for users and clients worldwide.
- **Fit Criterion:** The platform supports at least four major currencies (e.g., USD, EUR, GBP, INR) and correctly processes transactions in each.

## NFR-OE4

- **Priority:** Medium
- **Description:** The system shall be compatible with popular machine learning frameworks to facilitate the training, testing, and deployment of computer vision models.
- **Rationale:** Enables efficient development and integration of ML models for various use cases.

- **Fit Criterion:** The platform can train and deploy models using frameworks like TensorFlow, PyTorch, and scikit-learn without compatibility issues.

#### NFR-OE5

- **Priority: Low**
- **Description:** The system shall have efficient data pipelines for transferring labeled datasets between the platform and ML models.
- **Rationale:** Ensures smooth and automated workflows for data handling and model training.
- **Fit Criterion:** Data transfers occur within minutes, and pipelines support batch processing of large datasets (e.g., 10,000 images) without errors.

### 13.4 Productization Requirements

#### NFR-OE6

- **Priority: High**
- **Description:** The system shall be accessible without requiring any installation, operating solely through web browsers.
- **Rationale:** Eliminates the need for users to install software, reducing barriers to entry and simplifying access.
- **Fit Criterion:** Users can access and use the platform by visiting the web URL on any supported browser without needing to download or install additional software.

### 13.5 Release Requirements

#### NFR-OE7

- **Priority: Medium**
- **Description:** The system shall include a detailed release roadmap outlining timelines, features, and milestones for upcoming releases.

- **Rationale:** Keeps the development team and stakeholders aligned and ensures organized progress.
- **Fit Criterion:** The release roadmap is documented, updated regularly, and adhered to with at least 80% of milestones achieved on schedule.

#### NFR-OE8

- **Priority:** Low
- **Description:** The system shall be showcased to select group of users to gather feedback, identify bugs, and refine features before full-scale deployment.
- **Rationale:** Ensures the platform meets user expectations and operates smoothly upon official release.
- **Fit Criterion:** Testing results in fewer than 10 critical bugs identified and includes actionable feedback from at least 50 labellers, leading to necessary iterative improvements.

#### NFR-OE9

- **Priority:** Out of Scope
- **Description:** Each release shall preserve the core functionality of existing features without introducing regressions or deprecation.
- **Rationale:** Ensures continuity and reliability of the platform by preventing new updates from disrupting or breaking previously working features, thereby maintaining user trust and satisfaction.
- **Fit Criterion:** Automated usability testing is performed for each release, and no existing critical features fail post-release. Verification is achieved through passing all regression test cases before deployment.

## 14 Maintainability and Support Requirements

### 14.1 Maintenance Requirements

NFR-MR0

- **Priority:** Medium
- **Description:** All maintenance required for the system shall be possible to complete by a competent software developer after reading all the documentation provided in the source repository.
- **Rationale:** The system should be well documented, and therefore maintainable after reading said documents.
- **Fit Criterion:** A competent software developer, as determined by the original developers or their agents, can solve problems related to the core of the system, as determined by the features outlined in this document.

### 14.2 Supportability Requirements

N/A

### 14.3 Adaptability Requirements

N/A

## 15 Security Requirements

### 15.1 Access Requirements

NFR-SE0

- **Priority:** High
- **Description:** The application shall only allow users with labeling access, including labellers, customers, and admins, to view active projects and label images.

- **Rationale:** We do not want random users with no stake in the process to effect the results.
- **Fit Criterion:** Users who have not logged in to the application have no way of viewing projects or labeling images. Users logged in as labellers, customers, or admins have access to these features.
- **Related:** [FR10](#), [FR13](#)

#### NFR-SE1

- **Priority:** High
- **Description:** The application shall only allow users with customer access and above to create new image analysis projects.
- **Rationale:** Unidentified users creating projects would be impossible to facilitate. Also, labellers have no need to access project creation.
- **Fit Criterion:** Users who have not logged in to the application have no way of creating an image analysis project. Users logged in as customers or admins have access to these features.
- **Related:** [FR1](#), [FR4](#)

#### NFR-SE2

- **Priority:** High
- **Description:** The application shall validate the email format the user provides when creating an account.
- **Rationale:** We do not want users using invalid or duplicate emails to sign up.
- **Fit Criterion:** Let  $E$  represent the set of all email addresses, and let  $V$  represent the set of all valid email addresses. A valid email address conforms to the general pattern:

$V = (\forall \text{ email} \in E \mid \text{email matches the pattern } [a-zA-Z0-9+.-]+@[a-zA-Z0-9.-]+[a-zA-Z])$

New email  $\in V \wedge$  New email  $\neg \in$  Registered Emails

- **Related:** [FR0](#), [FR9](#)

### NFR-SE3

- **Priority:** High
- **Description:** The application shall validate the password format the user provides when creating an account.
- **Rationale:** We do not want users using weak passwords to sign up.
- **Fit Criterion:** Let  $P$  represent the set of all passwords, and let  $V$  represent the set of all valid passwords. A valid password has a at least one lowercase, uppercase, number and special character and is a minimum of 8 characters in length:

$$V = (\forall \text{ password} \in P \mid \text{password matches the pattern } (?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$\%&])[a-zA-Z0-9!@#$\%&]\{8,\})$$

- **Related:** [FR0](#), [FR9](#)

### NFR-SE4

- **Priority:** Medium
- **Description:** The system shall, during system failure, notify users trying to make a request of the system failure.
- **Rationale:** The system should provide good warning messages to users to prevent additional frustration.
- **Fit Criterion:** When services are unavailable, the system shall provide an error message to the user making the request of the unavailable service.
- **Related:** [FR8](#)

## 15.2 Integrity Requirements

### NFR-SE5

- **Priority:** Medium
- **Description:** The application shall prevent incorrect data from being introduced.
- **Rationale:** The database of information should always reflect correct and up to date information.
- **Fit Criterion:** The system must validate user inputs for data accuracy and format before they are saved. Any invalid data must trigger error messages, preventing it from being entered into the database. Users must be required to correct errors before proceeding.

### NFR-SE6

- **Priority:** Low
- **Description:** The application shall prohibit new user actions when user is disconnected from the system.
- **Rationale:** The system should avoid batch updates upon reconnect as to prevent malicious users creating a false package of updates.
- **Fit Criterion:** Upon user disconnect and reconnect, the system will not accept any updates from the user not submitted before the disconnect occurred.

### NFR-SE7

- **Priority:** Medium
- **Description:** The system shall reject duplicate entries in database.
- **Rationale:** Each entry to the database should be a unique data piece. If two entries are identical, they must differ by at least the creation time.
- **Fit Criterion:** No two entries in the database shall exist together for longer than 24 hours from their introduction into the database.

## 15.3 Privacy Requirements

### NFR-SE8

- **Priority:** Low
- **Description:** User data will be securely encrypted to protect user's privacy.
- **Rationale:** This will help to avoid user's being compromised if a data leak occurs.
- **Fit Criterion:** An encryption algorithm is used on sensitive user data such as passwords.

### NFR-SE9

- **Priority:** Out of Scope
- **Description:** The application shall ensure that all payment transactions are processed securely using encryption and comply with relevant security standards, such as PCI-DSS, which helps to protect payment account data (PCI Security Standards Council, 2024).
- **Rationale:** Protecting users' financial information is critical to maintaining trust. Failing to secure payments can lead to data breaches, financial loss, and legal liabilities.
- **Fit Criterion:** All payment transactions must use industry-standard encryption to protect sensitive data. Payment information, such as credit card details, must not be stored locally on the application and must be processed via a secure, PCI-DSS-compliant third-party payment gateway.
- **Related:** [FR3](#), [FR12](#)

## 15.4 Audit Requirements

N/A



## 15.5 Immunity Requirements

### NFR-SE10

- **Priority:** Medium
- **Description:** The application shall use parameterized queries or prepared statements for all database interactions.
- **Rationale:** We want to prevent SQL injection attacks which can lead to unauthorized data access or manipulation.
- **Fit Criterion:** All database queries must be implemented using parameterized queries or prepared statements. Dynamic SQL strings that concatenate user input must not be used in the codebase.

## 16 Cultural Requirements

### 16.1 Cultural Requirements

#### NFR-CU0

- **Priority:** Out of Scope
- **Description:** The system shall present users with the option to select the most popular language in each country it is deployed in.
- **Rationale:** It is important that the users of the program can understand what is said in each step.
- **Fit Criterion:** A drop down will allow users to select from the list of languages. At a minimum, the most popular language by number of speakers will be available for each country.

## 17 Compliance Requirements

### 17.1 Legal Requirements

#### NFR-CO0

- **Priority:** Low

- **Description:** The system shall not be available in any country currently facing economic sanctions by the Government of Canada.
- **Rationale:** Legal requirement to operate.
- **Fit Criterion:** Website must not be reachable in sanctioned countries. Canadian sanctions are outlined in the following legislation: [United Nation Act](#), [Special Economic Measures Act](#), and [Justice for Victims of Corrupt Foreign Officials Act](#)

#### NFR-CO1

- **Priority:** Low
- **Description:** The system shall follow Canadian tax code when accepting and paying out earnings.
- **Rationale:** Legal requirement to operate.
- **Fit Criterion:** All relevant tax codes must be satisfied when accepting payment from Customers or paying out earnings to Labellers. Specifically [Ontario Tax Act, 1990](#) and [Income Tax Act 1985](#)

#### NFR-CO2

- **Priority:** Low
- **Description:** The system shall allow Customers to restrict Labellers from certain regions from labeling the images related to their service request.
- **Rationale:** Customers may be uploading sensitive images, which are inappropriate for international Labellers to view.
- **Fit Criterion:** No restricted image, as identified by the customer, shall be shown to a restricted group.

## 17.2 Standards Compliance Requirements

N/A

## 18 Open Issues

**Task Assignment Algorithm:** To ensure labellers are engaged as they complete tasks and to obtain the highest quality of information possible, the system must implement an intelligent task allocation system. This system has not yet been determined.

**Label Consensus Algorithm:** Similarly, the algorithm for combining multiple user labels into one accurate label has not yet been determined.

**Labeling Services Offered:** The system has determined several potential labeling services to be offered by the system, but has not confirmed with certainty what will be included. This will be determined after more research has been completed on the task assignment algorithm.

## 19 Off-the-Shelf Solutions

### 19.1 Ready-Made Products

**Amazon Mechanical Turk:** A web-based crowdsourcing platform. Instead of building a novel front end, the system could obtain labels through this platform instead.

### 19.2 Reusable Components

**Label Studio:** A React library which contains components for building a web-based data annotation platform.

### 19.3 Products That Can Be Copied

**Toloka AI:** A general purpose image label crowdsourcing site. Supports image segmentation, bounding box drawing, and more computer vision labeling tasks. [Toloka.ai](https://toloka.ai)

## **20 New Problems**

### **20.1 Effects on the Current Environment**

The introduction of the OKKM Insights platform is expected to have several impacts on the existing technological and operational environment.

#### **20.1.1 Data Privacy and Security Concerns**

1. Impact: Handling sensitive satellite imagery and user-generated data raises significant privacy and security issues. Ensuring compliance with data protection regulations is most important.
2. Potential Problem: Breaches or mishandling of data could lead to legal repercussions, loss of trust, and reputational damage.

#### **20.1.2 Environmental Footprint**

1. Impact: Increased computational requirements for AI processing and data storage may lead to higher energy consumption.
2. Potential Problem: This could conflict with sustainability goals and lead to higher operational costs.

### **20.2 Effects on the Installed Systems**

Introducing the OKKM Insights platform will interact with and potentially disrupt existing systems within the organization and for stakeholders.

#### **20.2.1 Integration Challenges**

1. Impact: The platform will need to integrate with existing data sources, cloud services, and possibly applicable third-party APIs.
2. Potential Problem: Incompatibilities or integration failures could result in data inconsistencies, system downtimes, or increased maintenance efforts.

### **20.2.2 Legacy Systems Compatibility**

1. Impact: Older systems may not support the latest technologies required by the new platform.
2. Potential Problem: Upgrading or adjusting to legacy systems can be costly and time-consuming, potentially delaying the platform's deployment.

## **20.3 Potential User Problems**

Users are at the heart of the OKKM Insights platform, and several issues may arise that affect their experience and satisfaction.

### **20.3.1 Usability Issues**

1. Impact: If the platform is not intuitive or user-friendly, users may struggle to navigate and utilize its features effectively.
2. Potential Problem: Poor user experience can lead to reduced engagement, lower data labeling contributions, and higher dropout rates.

### **20.3.2 Training and Onboarding**

1. Impact: Users may require training to understand how to label data accurately and use the platform's tools.
2. Potential Problem: Inadequate training resources can result in inconsistent labeling, decreasing the quality of the datasets and the reliability of the AI models.

### **20.3.3 Compensation and Incentives**

1. Impact: Users expect fair compensation for their contributions.
2. Potential Problem: Delays or inaccuracies in payment processing can lead to dissatisfaction, reducing user retention and the overall quality of data labeling efforts.

#### **20.3.4 Technical Support**

1. Impact: Users may encounter technical issues that require timely resolution.
2. Potential Problem: Insufficient support can frustrate users, leading to decreased platform usage and negative word-of-mouth.

### **20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

Several environmental and contextual limitations could hinder the effective implementation and operation of the OKKM Insights platform.

#### **20.4.1 Regulatory Constraints**

1. Limitation: Different countries have varying regulations regarding satellite data usage, privacy, and AI applications.
2. Impact: Navigating these regulatory landscapes can be complex and may restrict the platform's operations in certain regions, limiting market potential.

#### **20.4.2 Technological Dependencies**

1. Limitation: The platform relies on third-party services (e.g., cloud providers, satellite data suppliers) whose availability and reliability are beyond the project's control.
2. Impact: Downtime or changes in third-party services can disrupt platform functionality and user experience.

### **20.5 Follow-Up Problems**

Addressing the initial set of problems may give rise to additional challenges that need to be managed.

### **20.5.1 Data Quality Management**

1. Follow-Up Problem: Ensuring the ongoing accuracy and relevance of labeled datasets as new data is continuously added. This may require implementing robust quality assurance mechanisms and periodic reviews.

### **20.5.2 User Engagement and Retention**

1. Follow-Up Problem: Continuously engaging users to maintain an active labeling workforce may require ongoing incentives, gamification strategies, and community-building efforts to prevent user fatigue and attrition.

### **20.5.3 Ethical Considerations**

1. Follow-Up Problem: Addressing ethical concerns related to surveillance, data misuse, and the potential dual-use nature of satellite imagery data. Establishing ethical guidelines and oversight mechanisms will be necessary to prevent misuse.

### **20.5.4 Dependency on User Participation**

1. Follow-Up Problem: The platform's success heavily relies on active user participation for data labeling. Changes in user engagement can directly impact data quality and availability, requiring strategies to stabilize user contributions.

### **20.5.5 Technical Debt Accumulation**

1. Follow-Up Problem: Rapid development to address emerging problems may lead to technical debt, where short-term solutions create long-term maintenance challenges. Proper code management and refactoring practices will be needed to mitigate this issue.

## 21 Tasks

### 21.1 Project Planning

- **Define Clear Milestones:** Establish milestones based on key project phases, such as requirements gathering, hazard analysis, proof of concept, backend development, and final project demonstration.
- **Task Assignment and Role Allocation:** Assign tasks to team members based on their strengths and expertise, with clear ownership over specific deliverables (e.g., backend development, frontend development, documentation, testing).
- **Use of Project Management Tools:** Utilize tools like Kanban boards in GitHub Projects to track progress, organize tasks into 'To Do,' 'In Progress,' and 'Completed' categories, and set internal deadlines to meet project milestones effectively.
- **Risk Management and Mitigation:** Identify potential project risks (e.g., data quality issues, development delays) and create mitigation strategies, including regular check-ins, backup plans, and prioritization of critical tasks.

### 21.2 Planning of the Development Phases

- **Requirements Gathering (24-Sep-2024 to 08-Oct-2024):** Engage in detailed discussions with stakeholders to gather comprehensive requirements for both frontend and backend components, ensuring all technical and business needs are documented.
- **Hazard Analysis (10-Oct-2024 to 23-Oct-2024):** Identify risks related to data handling, server security, and other critical project components. Develop a hazard analysis document to mitigate potential threats, especially in handling satellite imagery.
- **Verification & Validation (V&V) Planning (24-Oct-2024 to 01-Nov-2024):** Develop a detailed V&V plan, including test plans and quality assurance measures. This will include setting up testing environments and criteria for ensuring code correctness and reliability.



- **Proof of Concept (01-Nov-2024 to 22-Nov-2024):** Develop basic functionality for both backend and frontend, demonstrating image processing, API development, and the labeling workflow. Gather feedback to identify areas of improvement.
- **Backend Development (23-Nov-2024 to 05-Jan-2025):** Focus on developing API endpoints, integrating satellite image data, and ensuring secure and efficient handling of large datasets. Aim to have a functional backend by the end of this phase.
- **Frontend Development (23-Nov-2024 to 05-Jan-2025):** Design and implement UI/UX features, integrate backend APIs, and ensure an interactive user experience for both clients and labellers. Aim for a fully functional frontend that complements backend capabilities.
- **Design Document Revision (15-Dec-2024 to 15-Jan-2025):** Revise the design document to reflect final system architecture, covering database schema, API structure, and AI model integration.
- **Mid-Project Demonstration (03-Feb-2025 to 14-Feb-2025):** Prepare a functional prototype showcasing the integrated backend, frontend, and APIs. Demonstrate the labeling process, data flow, and key system functionalities.
- **Verification & Validation (15-Feb-2025 to 07-Mar-2025):** Test data processing pipelines, system reliability, and user interfaces to ensure accuracy and consistency. Run automated and manual tests for different use cases.
- **Final Project Demonstration & Expo Preparation (24-Mar-2025 to 01-Apr-2025):** Demonstrate the complete system, including all functionality, datasets, and models. Fine-tune UI and prepare for a polished presentation at the Expo.
- **Final Documentation (01-Apr-2025 to 15-Apr-2025):** Compile comprehensive final reports covering the entire development process, technical documentation, user manuals, and deployment guides.

## **22 Migration to the New Product**

### **22.1 Requirements for Migration to the New Product**

N/A

### **22.2 Data That Has to be Modified or Translated for the New System**

N/A

## **23 Costs**

Developing and implementing the requirements for the OKKM Insights platform involves various financial and effort-related expenditures. Below is a detailed breakdown of the primary cost components associated with building the platform:

### **23.1 Development Costs**

#### **23.1.1 Software Development**

1. **Front-End Development:** Creating a user-friendly web interface requires experience working with front-end technologies such as HTML, CSS, JavaScript, and frameworks like React or Angular.
2. **Back-End Development:** Developing robust server-side infrastructure using technologies like Node.js, Flask, and Python. This includes setting up databases, APIs, and integrating with cloud services (AWS/Azure).
3. **AI and Machine Learning Integration:** Building and integrating AI-powered features for automatic data labeling and computer vision model training will require specialized expertise in machine learning, potentially increasing development costs.

#### **23.1.2 UI/UX Design**

1. Investing time into UI/UX design to ensure the platform is intuitive and easy to use for both data labellers and end clients. This includes

designing workflows, views, and ensuring responsive design across devices.

### **23.1.3 Testing and Quality Assurance**

1. Comprehensive testing to identify and fix bugs, ensure cross-platform compatibility, and maintain high system reliability. This involves both automated and manual testing processes.

## **23.2 Infrastructure Costs**

### **23.2.1 Cloud Services**

1. Hosting: Utilizing cloud platforms like AWS or Azure for hosting the backend services and databases. Costs will scale with usage, including storage for satellite images, computational resources for AI processing, and data transfer fees.
2. Scalability: Ensuring the infrastructure can scale to handle increasing numbers of users and data volume, which may involve additional costs for load balancing, auto-scaling, and enhanced security measures.

### **23.2.2 Data Acquisition**

1. Purchasing high-quality, commercially available satellite images from third-party providers. These costs can vary based on the resolution, coverage area, and frequency of image updates required for different use cases.

## **23.3 Operational Costs**

### **23.3.1 User Compensation**

1. Labeling Incentives: Allocating funds to compensate users who contribute to the image labeling process. This includes setting competitive rates to attract and retain a large and active user base.
2. Payment Processing Fees: Costs associated with handling financial transactions, including fees from payment gateways for distributing earnings to users and receiving payments from clients.

### **23.3.2 Maintenance and Support**

1. Ongoing maintenance of the platform to ensure uptime, implement updates, and address technical issues. This also includes providing customer support to both data labellers and end clients.

### **23.3.3 Security and Compliance**

1. Implementing robust security measures to protect sensitive satellite data and financial transactions. Costs may include encryption technologies, regular security audits, and compliance with data protection regulations.

## **23.4 Marketing and User Acquisition Costs**

### **23.4.1 Promotional Activities**

1. Marketing efforts to attract both data labellers and end clients to the platform. This includes digital marketing campaigns, partnerships with relevant organizations, and participation in industry events.

### **23.4.2 Onboarding and Training**

1. Creating tutorials, documentation, and training materials to facilitate easy onboarding of new users and ensure they can effectively contribute to the labeling process with minimal friction.

## **23.5 Contingency and Miscellaneous Costs**

### **23.5.1 Unexpected Expenses**

1. Allocating a budget for unforeseen challenges such as technical setbacks, additional feature requests, or changes in market conditions that may require pivoting the project strategy.

## **23.6 Budget Forecast**

A detailed budget forecast will be developed, encompassing all the aforementioned cost categories. This forecast will be periodically reviewed and

adjusted based on project milestones, market conditions, and actual expenditure patterns to ensure financial sustainability and efficient resource allocation.

## 24 User Documentation and Training

### 24.1 User Documentation Requirements

#### NFR-UD0

- **Priority:** Out of Scope
- **Description:** A built-in help system with a searchable knowledge base, FAQs, tutorial videos, user forums and step-by-step guides to assist users with common issues or tasks directly within the platform.
- **Rationale:** An online help system provides quick, on-demand support, improving user experience by minimizing interruptions when seeking assistance.
- **Fit Criterion:** At least 80% of users should be able to resolve their issues using the help system without contacting support, as validated by user surveys and support metrics.

#### NFR-UD1

- **Priority:** Medium
- **Description:** A condensed guide offering a quick overview of key platform features and essential workflows to get users started rapidly.
- **Rationale:** A quick start guide enables new users to become productive immediately without needing to read extensive documentation, improving their onboarding experience.
- **Fit Criterion:** At least 80% of new users should report the quick start guide as helpful, and they should be able to complete a basic labeling task within 15 minutes of using it.

**NFR-UD2**

See [UH4](#)

## **24.2 Training Requirement**

**NFR-TR0**

See [UH4](#)

**NFR-TR1**

See [UH5](#)

## **25 Undesired Event Handling**

Please refer to Section 5 of the [Hazard Analysis](#) for a discussion on undesired event handling.

## **26 Waiting Room**

N/A

## **27 Likely Changes**

### **27.1 Likely Changes**

- Improvements to batch upload of images. It is likely at some point in the future of the project, users will prefer to upload multiple images to one project, and be able to view progress for each image from a single UI. This will require additional requirements to describe functionality.
- Payment support. When launched to customers, the site will have to support payment. This will require modifying the priority of existing requirements related to payments.
- Gamification of labelling. Currently, the labeller experience is a secondary priority in the project. To improve the experience for labellers,

we may add different types of labeling tasks, rewards, or additional elements to increase engagement.

## 28 Ideas for Solution

- **AI-Driven Labeling System:** Implement a machine learning model for preliminary image analysis to assist human labellers. This AI system can automatically pre-label sections of satellite images, allowing human labellers to validate or adjust these labels, significantly speeding up the process.
- **Gamification for Crowdsourced Labeling:** Introduce a gamification element to motivate and engage users who are labeling data. This can include earning points, badges, or rewards for completing labeling tasks accurately and efficiently, which can improve data quality and increase participation.
- **Intelligent Task Allocation:** Utilize an algorithm that matches labeling tasks with users based on their experience, preferences, and performance history. This ensures that difficult tasks are assigned to more experienced users while simpler tasks are made available to beginners.
- **Real-Time Collaboration and Consensus Model:** Allow multiple labellers to work on the same image in parallel, using a consensus model to determine the most accurate labels. Incorporate a review mechanism to resolve discrepancies and ensure that the final labeled data meets the required quality standards.
- **Automated Quality Assurance (QA) and Feedback:** Develop a QA system that automatically checks the quality of labeled data by comparing it to known benchmarks or running cross-validation with multiple labellers. Provide labellers with instant feedback to help improve their accuracy over time.
- **Modular Web-Based Platform Architecture:** Build a modular architecture for the platform, separating the frontend, backend, and AI components. This will allow easy updates, scalability, and the ability to add new features or services without disrupting existing functionality.

- **Flexible API for Client and User Integration:** Create an API that allows clients to easily upload satellite imagery, retrieve labeled data, and interact with the system programmatically. Also, provide endpoints for labellers to claim tasks, submit work, and check their progress or earnings.
- **Visualization Dashboard for Data Insights:** Provide clients with a dynamic dashboard that visualizes key insights derived from the labeled satellite imagery. This dashboard can display analytics, progress on labeling tasks, model accuracy, and any actionable insights relevant to the client's industry.
- **Secure Payment and Compensation System:** Implement a secure and transparent payment system that allows clients to pay for labeled data and compensates labellers based on their contributions. Use automated transaction management to ensure timely and accurate payments to all users.
- **Cross-Platform Compatibility and Accessibility:** Design the platform to be fully accessible across devices (desktop, mobile, tablets) and compliant with accessibility standards (e.g., WCAG). This ensures that a wide range of users, including those with disabilities, can participate in labeling and use the platform effectively.



## References

1. Government of Canada. (2017). *Government of Canada – Justice for Victims of Corrupt Foreign Officials Act*. <https://laws.justice.gc.ca/eng/acts/J-2.3/>
2. Government of Canada. (1992). *Government of Canada – Special Economic Measures Act*. <https://laws-lois.justice.gc.ca/eng/acts/S-14.5/index.html>
3. Government of Canada. (1985). *Government of Canada – Income Tax Act (1985)*. <https://laws-lois.justice.gc.ca/eng/acts/i-3.3/>
4. Government of Canada. (1985). *Government of Canada – United Nations Act*. <https://laws-lois.justice.gc.ca/eng/acts/U-2/index.html>
5. Government of Ontario. (1990). *Government of Ontario – Corporations Tax Act (1990)*. <https://www.ontario.ca/laws/statute/90c40>
6. PCI Security Standards Council. (2024, May 13). *PCI Security Standards Council – Protect Payment Data with Industry-driven Security Standards, Training, and Programs*. <https://www.pcisecuritystandards.org/standards/pci-dss/>
7. M Khaled. (2024). *The Attention Span Crisis in the Digital Age Explained—With Recent Statistics for 2024*. <https://profiletree.com/attention-span-crisis-digital-age-statistics/>
8. Hsueh, P.-Y & Melville, Prem & Sindhwani, Vikas. (2009). *Data quality from crowdsourcing: A study of annotation selection criteria*. Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing. 27-35.

## Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.
2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?

### Knowledge and Skills

#### Knowledge

- **Machine Learning Algorithms:** Understanding various machine learning algorithms, including supervised and unsupervised learning techniques, deep learning models, and their applications in satellite imagery analysis.
- **Model Evaluation and Tuning:** Gaining insights into model evaluation metrics (e.g., accuracy, precision, recall, F1 score) and techniques for hyperparameter tuning (e.g., grid search, random search, and Bayesian optimization) to improve model performance.

#### Skill

- **Data Handling and Preprocessing:** Developing skills in data preprocessing techniques such as data cleaning, normalization, augmentation, and handling missing values, which are crucial for preparing satellite imagery data for analysis.
- **Model Deployment:** Learning the practices of deploying machine learning models in production environments, including REST APIs and containerization using Docker.

## Approaches to Acquire Knowledge and Master Skills

- **Online Courses:**

- Enroll in comprehensive online courses focused on machine learning, such as:
  - \* “*Machine Learning*” by Andrew Ng (Coursera): Covers foundational concepts, algorithms, and applications of machine learning.
  - \* “*Deep Learning Specialization*” by Andrew Ng (Coursera): Provides in-depth knowledge of neural networks and their applications in image processing.
  - \* “*Machine Learning Crash Course*” by Google: A fast-paced, practical introduction to machine learning, featuring a series of lessons relating to ML models, data, and applications of ML.

- **Practical Projects:**

- Engage in hands-on projects that allow us to apply machine learning concepts to satellite imagery, such as:
  - \* Participating in *Kaggle competitions* that involve image classification or object detection in satellite images.
  - \* Collaborating with the team to develop a prototype of the labeling platform that incorporates real-time model predictions.

### Chosen Approach

I, Kartik, will primarily pursue **online courses** to build a strong theoretical foundation in machine learning, supplemented by **practical projects** to reinforce my learning through application. This combination enables me to understand complex machine learning concepts and apply them effectively to the capstone project, ensuring the team leverages cutting-edge techniques for analyzing satellite imagery.

I, Kyle, am currently enrolled in Fundamentals of Machine Learning, Data-Driven Algorithms for Sequential Decision-making, and Mathematical statistics. These courses will supplement the techniques Kartik described to help me learn the background knowledge required for the task allocation part

of our project. I foresee that being the biggest challenge, but I am very excited to work on it. While taking these courses, I also have access to the professors, who will likely be willing to discuss issues we run into while designing our algorithms.

I, Mathew, will focus on completing **online courses** to get a better understanding of the basics of machine learning. Specifically, I plan to follow google's machine learning crash course. This is because it combines video lectures with interactive visuals and practice exercises, which I think is an effective way to quickly gain knowledge on a topic.