

Software Requirements Specification for Software Engineering: subtitle describing software

Team #11, OKKM Insights

Mathew Petronilho

Oleg Glotov

Kyle McMaster

Kartik Chaudhari

October 8, 2024

Contents

1	Purpose of the Project	vii
1.1	Goals of the Project	vii
1.1.1	High Data Accuracy	vii
1.1.2	Ease of use	vii
1.1.3	Minimizing Cost to Analyze Images	vii
1.1.4	Results Returned Within Appropriate Timeframe	viii
1.1.5	High System Reliability and Accessibility	viii
2	Stakeholders	viii
2.1	Customer	viii
2.1.1	Users (Data Labelers)	viii
2.2	Other Stakeholders	ix
2.3	Hands-On Users of the Project	ix
2.4	Personas	ix
2.5	Priorities Assigned to Users	x
2.6	User Participation	x
2.7	Maintenance Users and Service Technicians	x
3	Mandated Constraints	x
3.1	Implementation Environment of the Current System	xi
3.2	Partner or Collaborative Applications	xi
3.3	Off-the-Shelf Software	xi
3.4	Anticipated Workplace Environment	xi
3.5	Schedule Constraints	xi
3.6	Budget Constraints	xii
4	Naming Conventions and Terminology	xii
4.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project	xii
5	The Scope of the Work	xiv
5.1	The Current Situation	xiv
5.2	The Context of the Work	xiv
5.3	Work Partitioning	xv
5.4	Specifying a Business Use Case (BUC)	xv

6	Business Data Model and Data Dictionary	xvii
6.1	Data Dictionary	xvii
7	The Scope of the Product	xviii
7.1	Product Boundary	xviii
7.2	Product Use Case Table	xix
7.3	Individual Product Use Cases (PUC's)	xxi
7.3.1	PUC 1 - Sign Up	xxi
7.3.2	PUC 2 - Log In	xxi
7.3.3	PUC 3 - Customer Log In	xxii
7.3.4	PUC 4 - Update User Information	xxii
7.3.5	PUC 5 - Annotate Satellite Images	xxii
7.3.6	PUC 6 - Segment Annotation	xxiii
7.3.7	PUC 7 - Bounding Box Annotation	xxiii
7.3.8	PUC 8 - Classification	xxiv
7.3.9	PUC 9 - Cash Out	xxiv
7.3.10	PUC 10 - View Image Insights	xxv
7.3.11	PUC 11 - Export Results	xxv
7.3.12	PUC 12 - Request Image Analysis	xxvi
7.3.13	PUC 13 - Upload Satellite Images	xxvi
7.3.14	PUC 14 - Request Satellite Images	xxvii
7.3.15	PUC 15 - Process Payment	xxvii
8	Functional Requirements	xxviii
8.1	Functional Requirements	xxviii
9	Look and Feel Requirements	xxxii
9.1	Appearance Requirements	xxxii
9.2	Style Requirements	xxxiii
10	Performance Requirements	xxxvii
10.1	Speed and Latency Requirements	xxxvii
10.2	Safety-Critical Requirements	xxxix
10.3	Precision or Accuracy Requirements	xxxix
10.4	Robustness or Fault-Tolerance Requirements	xxxix
10.5	Capacity Requirements	xl
10.6	Scalability or Extensibility Requirements	xl
10.7	Longevity Requirements	xl

11 Operational and Environmental Requirements	xli
11.1 Expected Physical Environment	xli
11.2 Wider Environment Requirements	xli
11.3 Requirements for Interfacing with Adjacent Systems	xli
11.4 Productization Requirements	xli
11.5 Release Requirements	xli
12 Maintainability and Support Requirements	xli
12.1 Maintenance Requirements	xli
12.2 Supportability Requirements	xlii
12.3 Adaptability Requirements	xlii
13 Security Requirements	xlii
13.1 Access Requirements	xlii
13.2 Integrity Requirements	xliii
13.3 Privacy Requirements	xliv
13.4 Audit Requirements	xliv
13.5 Immunity Requirements	xliv
14 Cultural Requirements	xliv
14.1 Cultural Requirements	xliv
15 Compliance Requirements	xlvi
15.1 Legal Requirements	xlvi
15.2 Standards Compliance Requirements	xlvi
16 Open Issues	xlvi
17 Off-the-Shelf Solutions	xlvi
17.1 Ready-Made Products	xlvi
17.2 Reusable Components	xlvi
17.3 Products That Can Be Copied	xlvi
18 New Problems	xlvi
18.1 Effects on the Current Environment	xlvi
18.1.1 Data Privacy and Security Concerns	xlvi
18.1.2 Environmental Footprint	xlvi
18.2 Effects on the Installed Systems	xlvi
18.2.1 Integration Challenges	xlvi

18.2.2	Legacy Systems Compatibility	xlix
18.3	Potential User Problems	xlix
18.3.1	Usability Issues	xlix
18.3.2	Training and Onboarding	xlix
18.3.3	Compensation and Incentives	xlix
18.3.4	Technical Support	l
18.4	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	l
18.4.1	Regulatory Constraints	l
18.4.2	Technological Dependencies	l
18.5	Follow-Up Problems	l
18.5.1	Data Quality Management	li
18.5.2	User Engagement and Retention	li
18.5.3	Ethical Considerations	li
18.5.4	Dependency on User Participation	li
18.5.5	Technical Debt Accumulation	li
19	Costs	liv
19.1	Development Costs	liv
19.1.1	Software Development	liv
19.1.2	UI/UX Design	liv
19.1.3	Testing and Quality Assurance	liv
19.2	Infrastructure Costs	lv
19.2.1	Cloud Services	lv
19.2.2	Data Acquisition	lv
19.3	Operational Costs	lv
19.3.1	User Compensation	lv
19.3.2	Maintenance and Support	lv
19.3.3	Security and Compliance	lvi
19.4	Marketing and User Acquisition Costs	lvi
19.4.1	Promotional Activities	lvi
19.4.2	Onboarding and Training	lvi
19.5	Contingency and Miscellaneous Costs	lvi
19.5.1	Unexpected Expenses	lvi
19.6	Budget Forecast	lvi

20 Waiting Room	lxi
20.1 Knowledge	lxiv
20.2 Skill	lxiv
21 Approaches to Acquire Knowledge and Master Skills	lxv
22 Chosen Approach	lxv

Revision History

Date	Version	Notes
10/1/2024	Mathew Petronilho	Added Purpose Of Project
Date 2	1.1	Notes

1 Purpose of the Project

There is currently a lack of high-quality, labeled satellite imagery datasets tailored for specific use cases. Many industries require specialized data for tasks like disaster response, environmental monitoring, urban planning, or defense, but building these datasets manually is time-consuming, costly, inefficient and may require expert data analysis. This hinders the development and deployment of accurate computer vision models for critical use cases across these various industries.

The purpose of this project is to create an online platform that accelerates this process and brings simplicity to satellite imagery data analysis.

1.1 Goals of the Project

1.1.1 High Data Accuracy

The system should have high classification accuracy for objects reported in the images. The core problem this system must solve is extracting useful information from the provided images. One key metric to determine the utility of the information found, is the classification accuracy of objects identified in the images. If the system is not able to determine what is contained in an image, it will not be useful to stakeholders.

1.1.2 Ease of use

The system should be very easy for stakeholders to use. There should be very low friction for users to classify images and objects found within images, with minimal training. It should also be simple for users to upload images to be analyzed. To maximize the information gained from users who are contributing to classification efforts, the system must ensure it is simple for users to get started with, and continue using the system. This is necessary to build a large enough user base, which will make it more likely to get insights in an acceptable amount of time.

1.1.3 Minimizing Cost to Analyze Images

The system should minimize the cost for users request insights from images. This could be implemented through intelligent algorithms for task delegation. Users of the system who upload images are interested in getting an

appropriate return for their investment. If the cost to analyze is too high, the platform will not retain a sufficiently large user base of purchasers.

1.1.4 Results Returned Within Appropriate Timeframe

The system should ensure the time it takes to obtain information from images is within a specified limit, as determined by users who upload images. Purchasers will have some time limit they require the system to process images within. To ensure timing needs are met, the system should provide realistic timelines and stick to them.

1.1.5 High System Reliability and Accessibility

The system should be useable remotely for purchasers and labellers, and have minimal downtime. The system should allow purchasers to upload images without being physically located where the system is hosted to ensure flexibility of use. The same should also be true for labellers, as they should be able to perform their tasks remotely. In both cases, the system should have low down time as to not introduce additional friction into the completion of tasks.

2 Stakeholders

2.1 Customer

End Clients/Customers: These stakeholders include governments, NGOs, private companies, and environmental organizations that pay for access to the labeled datasets and models. They rely on these datasets to make informed decisions in areas like environmental monitoring, urban planning, or defense-related tasks. Their satisfaction depends on the accuracy and reliability of both the data and the models provided.

2.1.1 Users (Data Labelers)

Data labelers are a core group of users responsible for annotating and classifying raw satellite imagery. Their role is fundamental to the project as they provide the labeled data required to train AI models. Data labelers are essential to the project's workflow, bridging the gap between raw data and

actionable insights for stakeholders. Their contributions not only shape the quality of the AI models but also impact the effectiveness of the end clients decision-making processes.

2.2 Other Stakeholders

Beyond the primary stakeholders, other key groups that benefit from high-quality satellite imagery datasets include:

- **Defense Agencies:** Rely on tailored data for surveillance, intelligence, and threat detection to enhance national security.
- **Environmental Agencies:** Use satellite data to monitor ecosystems, track deforestation, and respond to climate change.
- **Urban Planners:** Leverage data to manage land use, plan infrastructure development, and promote sustainable growth in cities.
- **Disaster Relief Organizations:** Depend on satellite imagery to assess damage in real-time and prioritize aid during crisis situations, making these datasets crucial for effective disaster response.
- **Image Labeling Teams:** Manually classify and annotate satellite images. Their work is crucial for building accurate datasets, and they benefit from improved tools and clearer guidelines to make the labeling process more efficient.

2.3 Hands-On Users of the Project

Users: These are individuals or entities responsible for labeling the data on the platform. In return for their efforts, they receive compensation. Their primary role is to ensure that the datasets are correctly annotated according to specified requirements, which forms the basis of the models developed. Their work directly impacts the quality and usability of the final product.

2.4 Personas

- **Air Rescue Analysts:** Use satellite imagery to assist in disaster management.

- **Financial Data Analysts:** Utilize satellite data for economic trend analysis.
- **Farmers/Agricultural Managers:** Monitor crop health, soil conditions, and weather for precision farming.

2.5 Priorities Assigned to Users

The importance of stakeholders can be prioritized based on their reliance on accurate and reliable datasets:

- **High Priority:** End Clients/Customers.
- **Medium Priority:** Air Rescue Services, Alternative Financial Data Companies.
- **Low Priority:** Urban Planners, Defense Agencies, Environmental Agencies.

2.6 User Participation

Users are integral to the platform as they label data in exchange for compensation. Their active participation is vital for ensuring that the datasets are annotated accurately and in accordance with the required specifications.

2.7 Maintenance Users and Service Technicians

These users are responsible for the upkeep of the platform, ensuring that the labeling tools and data processing pipelines are functioning smoothly. They also assist in troubleshooting any technical issues faced by the users.

3 Mandated Constraints

- 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.

3.1 Implementation Environment of the Current System

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

3.2 Partner or Collaborative Applications

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

3.3 Off-the-Shelf Software

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

3.4 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.

3.5 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](#).

3.6 Budget Constraints

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

4 Naming Conventions and Terminology

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

Term	Definition
Labelers	Users of our application that will be labeling photos shown to them for a monetary reward

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

- **Labelers 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 Labelers:** 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 labelers 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.

5 The Scope of the Work

5.1 The Current Situation

Insert your content here.

5.2 The Context of the Work

Insert your content here.

5.3 Work Partitioning

Insert your content here.

5.4 Specifying a Business Use Case (BUC)

Insert your content here.

6 Business Data Model and Data Dictionary

6.1 Data Dictionary

Data Object	Content	Type
Labeler	LabelerID	Class
Customer	CustomerID	Class
Labeler Payment Details	LabelerID Payment Details	Class
Customer Payment Details	CustomerID Payment Details	Class
Labeler Personal Details	LabelerID Personal Details	Class
Customer Personal Details	CustomerID Personal Details	Class
Historic Accuracy	LabelerID CategoryID Accuracy Samples	Class
Service Request	Image CustomerID Request Deadline	Class
Image	Image Data Image Metadata Labels	Class
Label	Labeled Class Bounding Box Labeler Label Metadata	Class
Object Class	Object Class List Object Class Score Labels Image	Class
Task Allocation	<div> xvii {Service Request +Available Labelers +Historic Accuracy +Object Class +Request Deadline } </div>	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	$a \in \mathbb{R} \wedge 0 \leq a \leq 1$	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

7 The Scope of the Product

7.1 Product Boundary

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

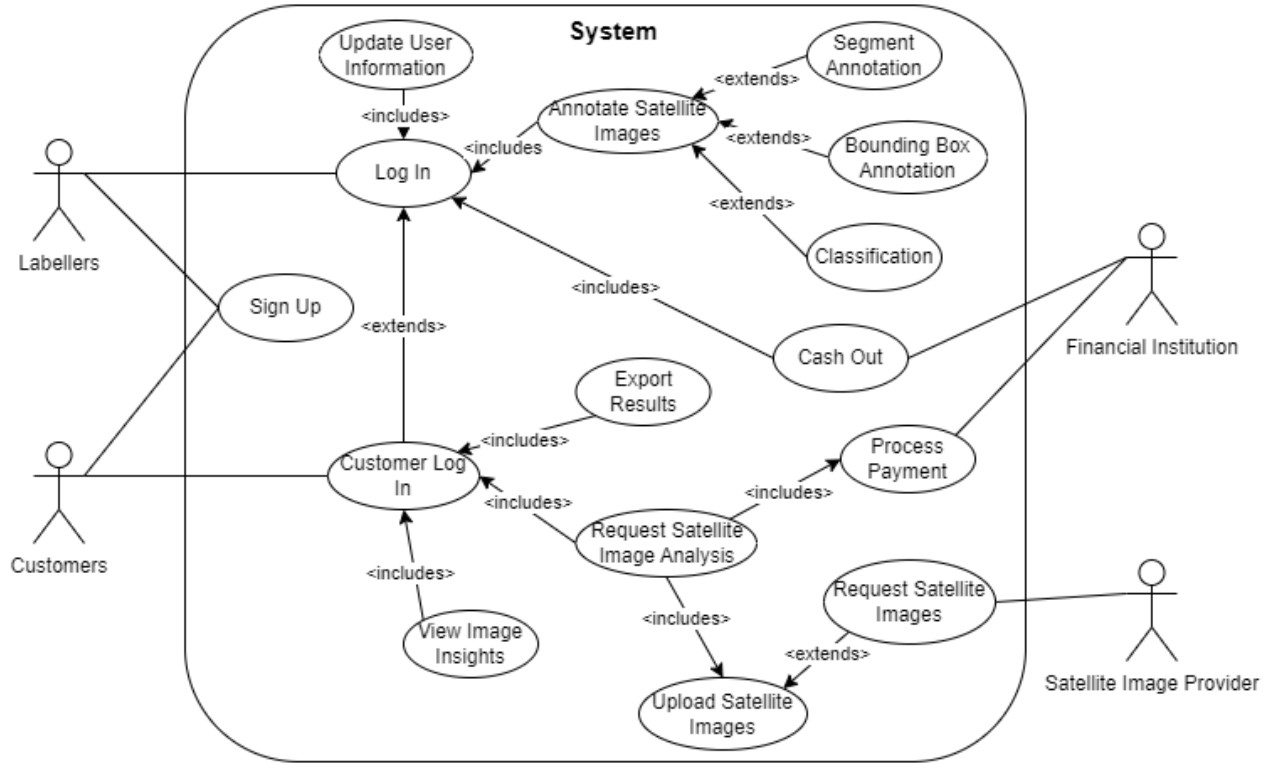


Figure 1: Use Case diagram

7.2 Product Use Case Table

PUC No	PUC Name	Actor/s	Input & Output
1	Sign Up	Labelers, Customers	Personal information & credentials (in), New account (out)
2	Log In	Labelers	Credentials (in)
3	Customer Log In	Customers	Credentials (in)
4	Update User Information	Labelers, Customers	Information to change (in), Updated information (out)

5	Annotate Images	Labelers	Image & annotation (in), Labeled image (out)
6	Segment Annotation	Labelers	Segmented image & user selections (in), Image with labeled segments (out)
7	Bounding Box Annotation	Labelers	Image & box annotations (in), Image with labeled boxes (out)
8	Classification	Labelers	Image & choices (in), Im- age labeled with the se- lected choice (out)
9	Cash Out	Labelers, 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 download- able 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, Fi- nancial Institu- tion	Financial information (in), Payment to app (out)

Table 2: Product Use Cases

7.3 Individual Product Use Cases (PUC's)

7.3.1 PUC 1 - Sign Up

Primary Actors: Labelers, 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

7.3.2 PUC 2 - Log In

Primary Actors: Labelers

Precondition: Labeler has signed up for an account

Trigger: Labeler selects log in

Success Scenario:

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

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

7.3.3 PUC 3 - 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 labeler

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

7.3.4 PUC 4 - Update User Information

Primary Actors: Labelers, 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

7.3.5 PUC 5 - Annotate Satellite Images

Primary Actors: Labelers

Precondition: Labeler has logged in

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

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

7.3.6 PUC 6 - Segment Annotation

Primary Actors: Labelers

Precondition: Labeler has logged in

Trigger: Labeler selects a listed project

Success Scenario:

1. The same as Annotate Images, but the labeler 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 labeler's reward balance is updated to reflect their submissions

7.3.7 PUC 7 - Bounding Box Annotation

Primary Actors: Labelers

Precondition: Labeler has logged in

Trigger: Labeler selects a listed project

Success Scenario:

1. The same as Annotate Images, but the labeler 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 labeler’s reward balance is updated to reflect their submissions

7.3.8 PUC 8 - Classification

Primary Actors: Labelers

Precondition: Labeler has logged in

Trigger: Labeler selects a listed project

Success Scenario:

1. The same as Annotate Images, but the labeler 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 labeler’s reward balance is updated to reflect their submissions

7.3.9 PUC 9 - Cash Out

Primary Actors: Labelers

Secondary Actor: Financial Institution

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

Trigger: Labeler selects cash out

Success Scenario:

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

Post-condition: The financial account that the labeler provided has gained money equal to the labeler’s reward balance

7.3.10 PUC 10 - 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

7.3.11 PUC 11 - 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

7.3.12 PUC 12 - 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 labelers
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 labelers so they can annotate the satellite images

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

7.3.13 PUC 13 - 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

7.3.14 PUC 14 - 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

7.3.15 PUC 15 - 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

8 Functional Requirements

8.1 Functional Requirements

FR0

- **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.

FR1

- **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.

FR2

- **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.

FR3

- **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.

FR4

- **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.

FR5

- **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.

FR6

- **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.

FR7

- **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.

FR8

- **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.

FR9

- **Description:** The system shall allow Labelers to create a new user account.
- **Rationale:** The Labelers must have an account to request the services provided by the system.
- **Fit Criterion:** Labeler is created in and stored in the database.

FR10

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

FR11

- **Description:** The system shall allow authenticated Labelers to edit their account information.
- **Rationale:** After creating an account, the Labelers must have the necessary tools to modify the information they have provided.
- **Fit Criterion:** After logging in, Labelers can modify the stored personal information.

FR12

- **Description:** The system shall allow Labelers to request their accrued earnings be transferred to their provided banking platform.
- **Rationale:** The system must ensure Labelers are compensated for their efforts.
- **Fit Criterion:** After logging in, Labelers can request their earnings be deposited in the account stored in their personal information.

FR13

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

FR14

- **Description:** The system shall combine annotations obtained from authenticated Labelers 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 (NFR-PR5).

9 Look and Feel Requirements

9.1 Appearance Requirements

Requirement LF1:

- **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.

Requirement LF2:

- **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.

9.2 Style Requirements

Requirement LF3:

- **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 Requirement

1. Clear Navigation Structure

- **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.

2. Consistent User Interface Elements

- **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

1. Task Prioritization Based on Preferences

- **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.

2. Localized Date, Time, and Currency Formats

- **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

1. In-App Progress Tracking for Learning

- **Description:** Users can track their progress through tutorials and training tasks within the platform, allowing them to see completed and pending learning modules.
- **Rationale:** Tracking progress motivates users to complete learning tasks and helps them understand where they stand in terms of mastering platform features.
- **Fit Criterion:** A progress tracker will be implemented, and at least 80% of users should find it helpful, based on post-training feedback surveys.

2. Interactive Practice Tasks

- **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

1. Contextual Help Pop-ups

- **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.

2. Friendly Error Handling and Guidance

- **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

1. Adjustable Text Size and Color Themes

- **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.

2. Keyboard Shortcuts for Core Actions

- **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.

10 Performance Requirements

10.1 Speed and Latency Requirements

NFR-PR0

- **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.
 $\text{IP}(t_{\text{newCustomerAccount}} < 0.25) \geq .90 \wedge \text{IP}(t_{\text{newCustomerAccount}} < 48) = 1$

NFR-PR1

- **Description:** The system shall process new user requests within 15 minutes of a Labeler completing the account creation process 90% of the time, and within 48 hours in all cases.
- **Rationale:** To improve engagement from Labelers, there should be minimal delay in getting started.
- **Fit Criterion:**
Let $t_{\text{newLabelAccount}}$ be the time it takes to process a new Labeler account request, in hours.
 $\text{IP}(t_{\text{newLabelAccount}} < 0.25) \geq 0.9 \wedge \text{IP}(t_{\text{newLabelAccount}} < 48) = 1$

NFR-PR2

- **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.
$$\mathbb{P}(t_{\text{serviceRequestTime}} < t_{\text{serviceRequestTimeLimit}}) \geq 0.9$$
$$\wedge \mathbb{P}(t_{\text{serviceRequestTime}} < t_{\text{serviceRequestTimeLimit}} + 48) = 1$$

NFR-PR3

- **Description:** The system shall take no longer than 10 seconds to display the next image to be labeled to a labeler, if there is one available.
- **Rationale:** To reduce friction for labelers, 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 [src](#).
- **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.
$$x_{\text{nextImage}} \Rightarrow t_{\text{imageServing}} \leq 10$$

NFR-PR4

- **Description:** The system shall deliver earned payouts to labelers within 7 business days of a request being made through the system.

- **Rationale:** The labelers 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.
 $t_{\text{payoutDelay}} < 7$

10.2 Safety-Critical Requirements

N/A

10.3 Precision or Accuracy Requirements

NFR-PR5

- **Description:** The system shall report accurate labels 75% of the time.
- **Rationale:** Average data label accuracy for competitors is greater than 75% ([link](#)). 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 .
 $(\forall o \in O : \mathbb{P}(L_{\text{True}}(o) = L_{\text{Guess}}(o)) \geq 0.75)$

10.4 Robustness or Fault-Tolerance Requirements

NFR-PR6

- **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_{uptime} be the uptime of the system.

Let t_{downtime} be the downtime of the system.

$$\frac{t_{\text{uptime}}}{t_{\text{uptime}} + t_{\text{downtime}}} > 0.97$$

10.5 Capacity Requirements

NFR-PR7

- **Description:** The system shall have the capacity to support enough labelers to meet all service request deadlines.
- **Rationale:** This requirement is critical to satisfy NFR-PR2.
- **Fit Criterion:** See NFR-PR2.

NFR-PR8

- **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 gigabytes in size.
- **Fit Criterion:** The system will not crash or fail to store when given images files 50 Gb in size.

10.6 Scalability or Extensibility Requirements

NFR-PR9

- **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.

10.7 Longevity Requirements

N/A

11 Operational and Environmental Requirements

11.1 Expected Physical Environment

Insert your content here.

11.2 Wider Environment Requirements

Insert your content here.

11.3 Requirements for Interfacing with Adjacent Systems

Insert your content here.

11.4 Productization Requirements

Insert your content here.

11.5 Release Requirements

Insert your content here.

12 Maintainability and Support Requirements

12.1 Maintenance Requirements

NFR-MR1

- **Description:** All maintainance required for the system shall be possible to complete by a competent software developer after reading all of 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.

12.2 Supportability Requirements

N/A

12.3 Adaptability Requirements

N/A

13 Security Requirements

13.1 Access Requirements

Requirement SR1:

- **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.

Requirement SR2:

- **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.

Requirement SR3:

- **Description:** The application shall validate the email format the user provides when creating an account.
- **Rationale:** We do not want users using invalid 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:

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

Requirement SR4:

- **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:

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

13.2 Integrity Requirements

Requirement SR5:

- **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.

13.3 Privacy Requirements

Requirement SR6:

- **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.

Requirement SR7:

- **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.

13.4 Audit Requirements

These requirements are not applicable as we are not an organization that is currently subject to audits.

13.5 Immunity Requirements

Requirement SR8:

- **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.

14 Cultural Requirements

14.1 Cultural Requirements

NFR-CUR1

- **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.

15 Compliance Requirements

15.1 Legal Requirements

NFR-COR1

- **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.

NFR-COR2

- **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 Labelers.

NFR-COR3

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

15.2 Standards Compliance Requirements

N/A

16 Open Issues

Task Assignment Algorithm: To ensure labelers 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.

17 Off-the-Shelf Solutions

17.1 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.

17.2 Reusable Components

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

17.3 Products That Can Be Copied

Tolka AI: A general purpose image label crowdsourcing site. Supports image segmentation, bounding box drawing, and more computer vision labeling tasks.

18 New Problems

18.1 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.

18.1.1 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.

18.1.2 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.

18.2 Effects on the Installed Systems

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

18.2.1 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.

18.2.2 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.

18.3 Potential User Problems

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

18.3.1 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.

18.3.2 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.

18.3.3 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.

18.3.4 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.

18.4 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.

18.4.1 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.

18.4.2 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.

18.5 Follow-Up Problems

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

18.5.1 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.

18.5.2 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.

18.5.3 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.

18.5.4 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.

18.5.5 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 labelers. 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.

19 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:

19.1 Development Costs

19.1.1 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.

19.1.2 UI/UX Design

1. Investing time into UI/UX design to ensure the platform is intuitive and easy to use for both data labelers and end clients. This includes designing workflows, views, and ensuring responsive design across devices.

19.1.3 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.

19.2 Infrastructure Costs

19.2.1 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.

19.2.2 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.

19.3 Operational Costs

19.3.1 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.

19.3.2 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 labelers and end clients.

19.3.3 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.

19.4 Marketing and User Acquisition Costs

19.4.1 Promotional Activities

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

19.4.2 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.

19.5 Contingency and Miscellaneous Costs

19.5.1 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.

19.6 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.

23 User Documentation and Training

23.1 User Documentation Requirements

1. Comprehensive User Manual

- **Description:** A complete user manual detailing all platform functionalities, including system access, navigation, image labeling, and account management, enhanced with visual aids like screenshots for clarity.
- **Rationale:** A thorough manual helps users find information independently, reducing support queries and ensuring they can maximize the platform's potential.
- **Fit Criterion:** The manual should cover at least 95% of the platform's features, and user feedback should reflect a 90% satisfaction rate in terms of usefulness and comprehensiveness.

2. Online Help System

- **Description:** A built-in help system with a searchable knowledge base, FAQs, 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.

3. API Documentation for Developers

- **Description:** Detailed API documentation using a standard like OpenAPI, covering all endpoints, parameters, response formats, and authentication mechanisms to facilitate integration by external developers.
- **Rationale:** Well-documented APIs empower developers to integrate their applications with the platform seamlessly, fostering a wider ecosystem and usage.

- **Fit Criterion:** The API documentation should be comprehensive and accurate, with at least 90% of developer feedback confirming clarity and ease of use.

4. Release Notes and Updates

- **Description:** Timely release notes detailing new features, bug fixes, and system changes, ensuring users are kept informed about platform improvements and updates.
- **Rationale:** Regular release notes maintain transparency, keeping users informed and reducing confusion about new functionality or changes to existing features.
- **Fit Criterion:** Each major platform update should be accompanied by release notes, with at least 95% coverage of changes, and users should report a clear understanding of new features based on feedback.

5. Quick Start Guide

- **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 90% 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.

6. Contextual In-App Help

- **Description:** Provide contextual help directly within the platform, such as tooltips and small info pop-ups, offering real-time guidance for features or form fields as users interact with them.
- **Rationale:** In-app help reduces the learning curve and provides timely support without requiring users to leave the page or workflow they are on.

- **Fit Criterion:** At least 85% of users should find in-app help intuitive and accurate in providing the necessary support as validated by user surveys.

23.2 Training Requirement

1. Interactive Onboarding Tutorial

- **Description:** An onboarding tutorial guiding users through key platform functions interactively, offering tooltips and prompts to familiarize them with core workflows.
- **Rationale:** Interactive onboarding accelerates learning, reduces user frustration, and ensures that new users can quickly perform tasks confidently.
- **Fit Criterion:** At least 85% of users should complete the onboarding tutorial successfully within 20 minutes, with a 90% rate of positive feedback on its effectiveness.

2. Video Tutorials and Webinars

- **Description:** Short, focused video tutorials covering specific platform tasks (e.g., labeling, account setup) and periodic webinars for in-depth learning and live Q&A sessions.
- **Rationale:** Video content provides a visual and engaging method for users to learn, while webinars offer opportunities for real-time interaction and deeper understanding.
- **Fit Criterion:** Each key platform feature should have a corresponding video tutorial, and at least 90% of webinar attendees should rate the sessions as helpful in post-event feedback.

3. Training for Specific User Groups

- **Description:** Targeted training materials and sessions tailored to different user groups, such as labelers, clients, and administrators, focusing on their unique needs and platform usage.
- **Rationale:** Customized training ensures that users receive relevant guidance, making them more efficient and effective in their roles on the platform.

- **Fit Criterion:** At least 90% of users in specific groups should report that their training materials are relevant to their needs and improve their platform experience.

4. Self-Assessment and Practice Environment

- **Description:** A sandbox environment allowing users to practice labeling tasks without affecting live data, including quizzes and exercises to assess understanding and improve skills.
- **Rationale:** A practice environment enables users to build confidence and hone their skills in a low-pressure setting, ensuring they are prepared for real tasks.
- **Fit Criterion:** At least 80% of new users should utilize the practice environment, with self-assessment scores indicating an average improvement of 20% in labeling accuracy over their first three attempts.

5. Ongoing Support and Community Forum

- **Description:** Provide ongoing support through a community forum or discussion board where users can ask questions, share experiences, and learn from one another.
- **Rationale:** A support forum fosters a sense of community, provides peer-to-peer assistance, and reduces the load on formal support channels.
- **Fit Criterion:** At least 75% of user questions in the community forum should be answered by either other users or support staff within 2 days.

6. Periodic User Training Assessments

- **Description:** Conduct periodic assessments of user knowledge through quizzes or practical tasks to identify training gaps and offer refresher sessions where needed.
- **Rationale:** Regular assessments help ensure users retain knowledge and stay up-to-date with platform changes, enhancing overall productivity and satisfaction.

- **Fit Criterion:** At least 70% of users should participate in training assessments, with scores indicating a consistent understanding of platform features and workflows.

20 Waiting Room

Insert your content here.

25 Ideas for Solution

- **AI-Driven Labeling System:** Implement a machine learning model for preliminary image analysis to assist human labelers. This AI system can automatically pre-label sections of satellite images, allowing human labelers 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 labelers 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 labelers. Provide labelers 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 labelers 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 labelers 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. 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/>

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?

sectionKnowledge and Skills

20.1 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.

20.2 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.

21 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.
- **Practical Projects:**
 - Engage in hands-on projects that allow me 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.

22 Chosen Approach

I 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.