



CSC 431

EcoDex

Software Requirements Specification (SRS)

Team 03

Michael Castellucci	Scrum Master
Carter Falkenberg	Developer
Phuong Uyen Dang	Developer

Version History

Version	Date	Author(s)	Change Comments
1.0	02/20/24	Michael Castellucci Carter Falkenberg Phuong Uyen Dang	First draft
1.1	05/06/24	Michael Castellucci, Phuong Uyen Dang Carter Falkenberg	Fixes made to first draft

Table of Contents

1.	System Requirements	
1.1	Functional Requirements	7
1.1.1	Log Flora/Fauna	7
1.1.2	Identify Flora/Fauna	7
1.1.3	Save User Entry	7
1.1.4	Make Account	7
1.1.5	View Logs Map	8
1.1.6	Add Friends	8
1.1.7	Login to Account	8
1.1.8	View Friend Feed	8
1.1.9	Send Message	9
1.1.10	Report User	9
1.1.11	Get Badges	9
1.1.12	Adjust Settings	9
1.1.13	View Profile	10
1.1.14	Remove Friends	10
1.2	Non-Functional Requirements	10
1.2.1	Classify Flora/Fauna in less than 1 second	10
1.2.2	Encrypt User Information	10
1.2.3	Portable on IOS and Android	10
1.2.4	Scalable up to 100,000 users	11
1.2.5	Community Guidelines	11
2.	System Constraints	
2.1	Tool Constraints	12
2.1.1	Source Control	12
2.1.2	Mobile App Framework	12
2.1.3	Backend Framework	12
2.1.4	Server	12
2.2	Language Constraints	12
2.2.1	Frontend	12
2.2.2	Backend	13
2.2.3	Classification Model	13
2.3	Platform Constraints	13
2.3.1	Mobile	13
2.4	Network Constraints	13
2.4.1	Internet Access	13
2.4.2	Internet Speed	13
2.5	Transition & Support Constraints	14
2.5.1	End of Life	14
2.6	Budget & Schedule Constraints	14
2.6.1	First Deployable Version Time Constraint	14
2.6.2	Budget Constraint	14
2.7	Storage Constraints	14
2.7.1	User Data	14
2.7.2	Map Data	14
2.7.3	Model Parameters	15
3.	Requirements Modeling	
3.1.1	System Use Case Diagram	16

4.	Evolutionary Requirements	
4.1	Functional Requirements	17
4.1.1	Create Group	17
4.1.2	Manage Group	17
4.1.3	Make Post	17
4.1.4	Comment on Post	17
4.1.5	React to Post	18
4.1.6	Flora/Fauna of the Day	18
4.1.7	View Account History	18
4.1.8	Block/Unblock User	18
4.2	Non-Functional Requirements	19
4.2.1	Automated Recording of Account Events	19
4.2.2	Automated Detection of Suspicious Activity	19

Table of Tables

1. System Requirements

Table 1: Log Flora/Fauna	7
Table 2: Identify Flora/Fauna	7
Table 3: Save User Entry	7
Table 4: Make Account	8
Table 5: View Logs Map	8
Table 6: Add/Remove Friends.....	8
Table 7: Login to Account.....	8
Table 8: Friend Feed.....	8
Table 9: Chat	9
Table 10: Report User	9
Table 11: Get Badges	9
Table 12: Settings	9
Table 13: Classify Flora/Fauna in less than 1 second	10
Table 14: Encrypt User Information	10
Table 15: Portable on IOS and Android	10
Table 16: Scalable up to 100,000 users	10
Table 17: Community Guidelines	10

2. System Constraints

Table 18: Source Control	11
Table 19: Mobile App Framework	11
Table 20: Backend Framework	11
Table 21: Server	11
Table 22: Frontend	11
Table 23: Backend	12
Table 24: Classification Model	12
Table 25: Mobile	12
Table 26: Internet Access	12
Table 27: Internet Speed	12
Table 28: End of Life	13
Table 29: First Deployable Version Time Constraint	13
Table 30: Budget Constraint	13
Table 31: User Data	13
Table 32: Map Data	13
Table 33: Model Parameters	14

4. Evolutionary Requirements

Table 34: Create Group	16
Table 35: Manage Group	16
Table 36: Make Post	16
Table 37: Comment on Post	16
Table 38: React to Post	17
Table 39: Flora/Fauna of the Day	17
Table 40: View Account History	17
Table 41: Block/Unblock User	17
Table 42: Record Account Event	18
Table 43: Report Suspicious Account Activity	18

Table of Figures

Figure 1: System Use Case Diagram	15
---	----

1. System Requirements

1.1 Functional Requirements

1.1.1 Log Flora/Fauna

Title	Log flora/fauna
Description	The user can take a picture of a plant/animal/fungi and upload it to the app for identification
Priority	0
Precondition(s)	User must be logged into an account
Basic Flow	The user makes a log by inputting a picture of flora/fauna, as well as an optional comment, location, and the user is given a classification recommendation
Postconditions(s)	Log is added to the observations map and shown on friends feed. Log is included in database
Use Case Diagram	3.1.1

1.1.2 Identify Flora/Fauna

Title	Identify flora/fauna
Description	Image classification of plant/animal/fungi
Priority	0
Precondition(s)	User must take photo of plant/animal/fungi
Basic Flow	The user inputs a photo into our app to get a classification of the organism without making a log
Postconditions(s)	The user is sent back to the home page
Use Case Diagram	3.1.1

1.1.3 Save User Entry

Title	Save user entry
Description	After classification, log the photo, classification, and location in the user's database.
Priority	0
Precondition(s)	The user must submit a photo for classification, user must be logged in
Basic Flow	After the user takes a photo that is classified, the metadata is automatically saved to the user's account.
Postconditions(s)	The user is sent back to the home page
Use Case Diagram	3.1.1

1.1.4 Make Account

Title	Make account
Description	The user creates an EcoDex account to keep track of their logs, badges, friends, etc.

Priority	0
Precondition(s)	
Basic Flow	The first page encountered is a login page with an option to create an account. When the creates an account, the account creation page takes email, name, and username and creates an account with it.
Postconditions(s)	Allows access to friend management and adding logs
Use Case Diagram	3.1.1

1.1.5 View Logs Map

Title	View observations map
Description	User can view other users' observations around the world
Priority	2
Precondition(s)	User must be logged into an account
Basic Flow	User views a page that displays a world map with pins where other users have made logs
Postconditions(s)	Logs can be viewed in detail
Use Case Diagram	3.1.1

1.1.6 Add Friends

Title	Add friend
Description	Users can connect with each other by username
Priority	2
Precondition(s)	User must be logged into an account
Basic Flow	Users choose friends to add. This will allow them to see their logs in the friend feed.
Postconditions(s)	Users can see each others logs as they are created
Use Case Diagram	3.1.1

1.1.7 Login to Account

Title	Login to account
Description	User is able to access their created account
Priority	1
Precondition(s)	User must have an active account
Basic Flow	The user inputs their account information when opening EcoDex to access their account.
Postconditions(s)	Access to all other features of EcoDex
Use Case Diagram	3.1.1

1.1.8 View Friend Feed

Title	Friend Feed
Description	Page where the user can see their friends' logs
Priority	2
Precondition(s)	User must be logged in and have friends
Basic Flow	The user views recent logs of all their friends that are collected on one interface.
Postconditions(s)	User is able to interact with posts or go back to home page

Use Case Diagram	3.1.1
------------------	-------

1.1.9 Send Message

Title	Send Message
Description	Tab where user can message with another user
Priority	2
Precondition(s)	User must be logged in and have friends
Basic Flow	The user directly messages with any of their friends through EcoDex.
Postconditions(s)	Message is displayed to a friend, and the user can see replies
Use Case Diagram	3.1.1

1.1.10 Report User

Title	Report User
Description	An option in each user's profile to report
Priority	2
Precondition(s)	User must be logged into an account
Basic Flow	The user reports another user's profile for inappropriate behavior after viewing their account.
Postconditions(s)	User can specify the reason(s) for reporting
Use Case Diagram	3.1.1

1.1.11 View Badges

Title	Get Badges
Description	Badges are given based on specifics of a log. i.e. if a rare animal is spotted
Priority	3
Precondition(s)	User must be logged into an account
Basic Flow	The user views all the badges they have earned sorted by location and other variables like diet, habitat, range, etc.
Postconditions(s)	User can view what badges they've just unlocked and return to home screen
Use Case Diagram	3.1.1

1.1.12 Adjust Settings

Title	Adjust Settings
Description	User can adjust account settings
Priority	3
Precondition(s)	User must be logged in
Basic Flow	The user views their profile and clicks on the Gear icon to view and adjust a variety of settings like username, password, UI themes, and more.
Postconditions(s)	Users return to the home screen with their new settings
Use Case Diagram	3.1.1

1.1.13 View Profile

Title	Settings
Description	User can view the profile page
Priority	3
Precondition(s)	User must be logged in
Basic Flow	The user views the profile page and sees their username, bio, badges, and settings icon.
Postconditions(s)	User can click on the Gear icon to view and adjust their account settings
Use Case Diagram	3.1.1

1.1.14 Remove Friends

Title	Remove friends
Description	Users are able to connect with each other by username
Priority	2
Precondition(s)	User must be logged in and have a friend to remove
Basic Flow	User chooses friends to remove. This will prevent their logs from showing on the user's friend feed.
Postconditions(s)	User is given a prompt saying the friend has been removed
Use Case Diagram	3.1.1

1.2 Non-Functional Requirements

1.2.1 Classify Flora/Fauna in less than 1 second

Title	Classify flora/fauna in less than 1 second
Description	When the user takes a photo, the plant/animal/fungi should be classified within 1 second
Priority	3
Applicable FR(s)	1.1.2

1.2.2 Encrypt User Information

Title	Encrypt user information
Description	The user's data should all be secured and encrypted. This includes name, email, password, friends, location data, and photos.
Priority	1
Applicable FR(s)	1.1.1, 1.1.3, 1.1.4, 1.1.7, 1,1,12

1.2.3 Portable on IOS and Android

Title	Portable on IOS and Android
Description	The app should work for both IOS and Android users equally.

Priority	0
Applicable FR(s)	1.1.1-1.1.14

1.2.4 Scalable up to 100,000 users

Title	Scalable up to 100,000 users
Description	Until nearly 100,000 users are active, the app should be completely functional for up to 100,000 users. Speed, reliability, and robustness should not be affected.
Priority	2
Applicable FR(s)	1.1.3, 1.1.4

1.2.5 Community Guidelines

Title	Community Guidelines
Description	Ensure user interactions stay civil and within terms defined in a page about Community Guidelines to protect users from possible negative community experiences.
Priority	1
Applicable FR(s)	1.1.4, 1.1.9, 1.1.10

2. System Constraints

2.1 Tool Constraints

2.1.1 Source Control

Title	Source Control
Description	GitHub will be used for source control. Source: https://github.com/
Priority	0

2.1.2 Mobile App Framework

Title	Mobile App Framework
Description	React native will be used so that the app can be deployed to both Android and IOS Source: https://reactnative.dev/
Priority	0

2.1.3 Backend Framework

Title	Backend Framework
Description	We will use django and pyinaturalist, written in python, in order to make calls to the classification model. Source: https://github.com/pyinat/pyinaturalist Source: https://www.diangoproject.com/
Priority	0

2.1.4 Server

Title	Server
Description	We will host a server using Microsoft Azure https://azure.microsoft.com/en-us
Priority	0

2.2 Language Constraints

2.2.1 Frontend

Title	Frontend
Description	React native will be used for the front end.
Priority	0

2.2.2 Backend

Title	Backend
Description	Python, specifically Django, will be used for the backend. We will be making calls to INaturalist API via pyinaturalist on github. Source: https://github.com/pyinat/pyinaturalist Source: https://www.djangoproject.com/
Priority	0

2.2.3 Classification Model

Title	Classification
Description	The classification model will be written in python, utilizing INaturalist's python API for source images and labels. Source: https://github.com/pyinat/pyinaturalist
Priority	0

2.3 Platform Constraints

2.3.1 Mobile

Title	Mobile
Description	The application will be available only on Android and IOS mobile devices, in each of the respective app stores.
Priority	0

2.4 Network Constraints

2.4.1 Internet Access

Title	Internet Access
Description	The user must be connected to the internet in order to use the application.
Priority	0

2.4.2 Internet Speed

Title	Internet Speed
Description	The user must have a minimum internet speed of 10Mbps upload and 100 Mbps download so that photos can be uploaded to and downloaded from the server quickly.
Priority	5

2.5 Transition & Support Constraints

2.5.1 End of Life

Title	End of Life
Description	At the end of this class (Introduction to Software Engineering), all support for this project will stop indefinitely.
Priority	5

2.6 Budget & Schedule Constraints

2.6.1 First Deployable Version Time Constraint

Title	First Deployable Version Time Constraint
Description	The first deployable version of the application should be available by May 1, 2024 for both IOS and Android.
Priority	1

2.6.2 Budget Constraint

Title	Budget Constraint
Description	No budget can be spent on the development of this project. Only free or trial softwares should be employed.
Priority	1

2.7 Storage Constraints

2.7.1 User Data

Title	User Data
Description	Stored in Azure. Email, username, password, images and their metadata (time, location, classification), friends list, and user settings preferences (location services, data saver mode).
Priority	2

2.7.2 Map Data

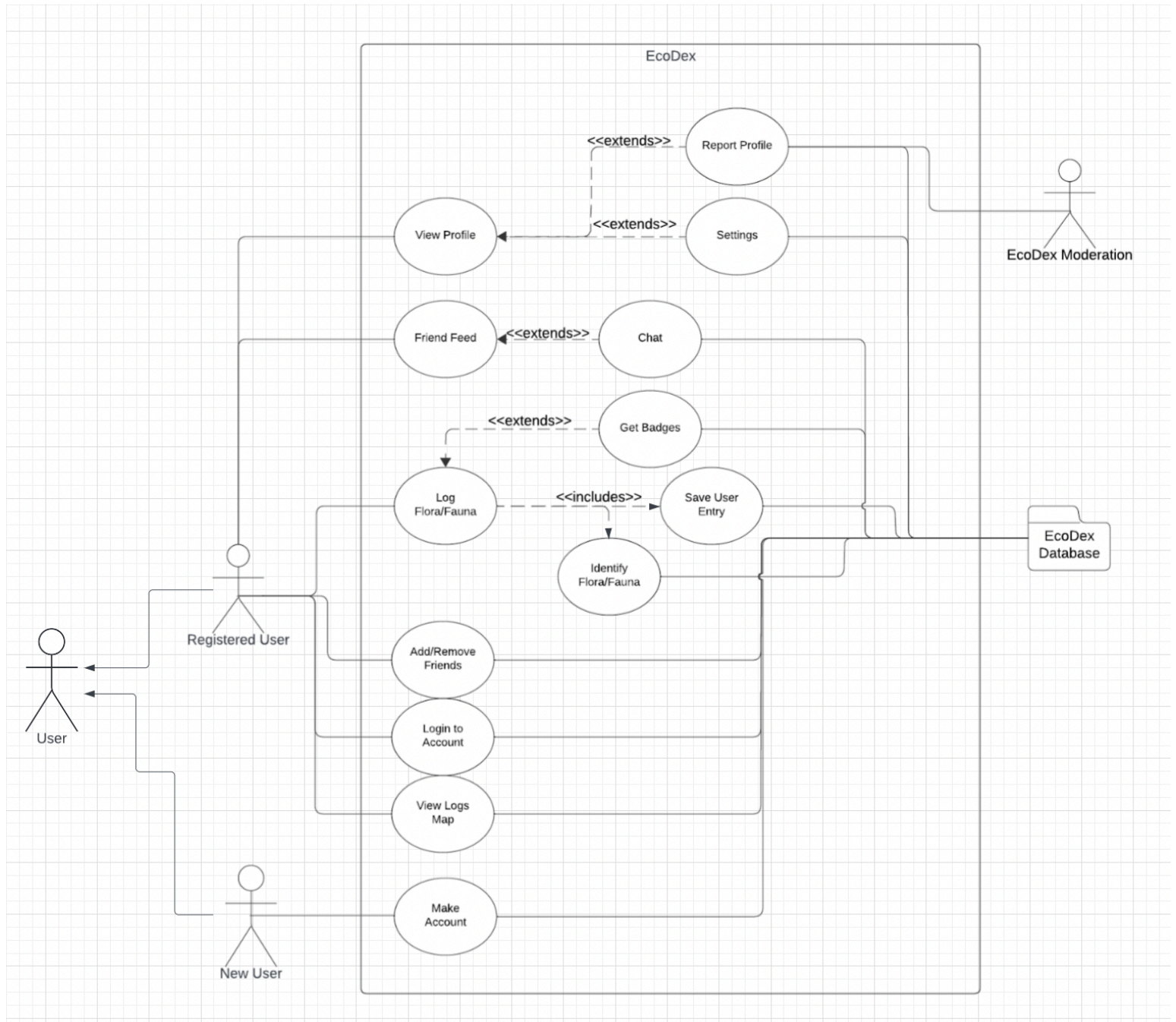
Title	Map Data
Description	Stored in Azure. Includes fields such as location names and other relevant information about locations such as geography, local flora/fauna, etc.
Priority	1

2.7.3 Model Parameters

Title	Model Parameters
Description	Stored in the server, in Azure. Model must not be too large to be stored in the server. Server must have enough space to store model parameters.
Priority	0

3. Requirements Modeling

3.1.1 System Use Case Diagram



4. Evolutionary Requirements

4.1 Functional Requirements

4.1.1 Create Group

Title	Create Group
Description	Users can join together in groups where they can share their finds and earn badges together.
Priority	5
Precondition(s)	The user must have an account and friends.
Postconditions(s)	Allows access to group functions and activities.
Use Case Diagram	N/A

4.1.2 Manage Group

Title	Manage Group
Description	A user designated as the group leader is able to invite, kick, and mute members in their group.
Priority	5
Precondition(s)	A group must be created with members.
Postconditions(s)	Group homepage is shown along with group posts (if any).
Use Case Diagram	N/A

4.1.3 Make Post

Title	Make Post
Description	Users have the option to make a post separate from a log, where they are allowed more freedom in what they post and talk about
Priority	5
Precondition(s)	User needs an account.
Postconditions(s)	Post will be displayed to friends of that user and the user will be brought back to home page.
Use Case Diagram	N/A

4.1.4 Comment on Post

Title	Comment on Post
Description	Users can write text comments on their friends' posts.
Priority	5
Precondition(s)	User must have friends and one of them must have posted.
Postconditions(s)	Friends can see their comments and then, the user is brought back to post view.
Use Case Diagram	N/A

4.1.5 React to Post

Title	React to Post
Description	Works in the same way as 4.1.4, but with a single emoji.
Priority	5
Precondition(s)	Same as 4.1.4
Postconditions(s)	Same as 4.1.4
Use Case Diagram	N/A

4.1.6 Flora/Fauna of the Day

Title	Flora/Fauna of the Day
Description	The system automatically finds a fascinating flora/fauna to display it to all users. This action is repeated daily.
Priority	5
Precondition(s)	User navigates to home page to view flora/fauna of the day
Postconditions(s)	An organism of the day is displayed to the user along with some facts about it.
Use Case Diagram	N/A

4.1.7 View Account History

Title	View Account History
Description	Account history could be viewed by the user as a timeline of log entries, unlocked species, badges, and other significant account events.
Priority	4
Precondition(s)	The user navigates to their profile and chooses the option to View Account History.
Postconditions(s)	A timeline of the account history is visible to the user. The user can interact with the events on the timeline by clicking on any event to view more information including location and time (if applicable).
Use Case Diagram	N/A

4.1.8 Block/Unblock User

Title	Block/Unblock User
Description	The user is able to view and block/unblock other users from their profiles.
Priority	4
Precondition(s)	The user must be able to view another user's profile.
Postconditions(s)	Another user is thus blocked and cannot interact with the user, including viewing the user's feed or chatting with the user, and vice versa for the user. If the blocked account is friends with the user, this action automatically unfriends

	them. The user can still view a blocked account's profile and choose to unblock it.
Use Case Diagram	N/A

4.2 Non-Functional Requirements

4.2.1 Automated Recording of Account Events

Title	Automated Recording of Account Events
Description	System automatically records the location and time (if applicable) of certain account events for a summary of the account history which could then be viewed from the user's profile.
Priority	4
Applicable FR(s)	4.1.7

4.2.2 Automated Detection of Suspicious Activity

Title	Automated Detection of Suspicious Activity
Description	System automatically recognizes activities (such as important account information changes) that could not be reasonably attributed to the rightful account owner and sends a warning email to the email address currently listed in account information.
Priority	4
Applicable FR(s)	All