Software Requirements Specification

For

By

Millennium

Name	Student Number
Charl Pretorius	u22519042
Ivan Horak	u21456552
Matthew Botha	
Justin Hudson	u21543152
Jarrod Jefferies	u18003193

Contents

Introduction:

BeakPeek aims to grow the bird watching community in South Africa by making it easier to find, identify and predict the occurrence of bird species.

BeakPeek will allow users to see what bird species can be found in their area and provide conservationists with an analytical overview of the data to easily gauge the growth and decline of local bird species.

User Characteristics:

User 1 – Birders: The primary user of the application, birders can use the application to view all kinds of birds in South Africa and where to find them using SABAP2's public database. Birders can view what birds can be seen in the pentad they are currently in. They can view the list of birds that they have seen and add their sighting to the database if they want to. Birders can identify birds by taking a photo of it. Birders can identify birds through specialised filters. They can also listen to what a bird sounds like.

User 2 – Conservationists: Conservationists can use the data uploaded by users and by SABAP2 to receive an indication of what a specific bird population is doing, more sightings than expected could indicate an increase in the birds population and vise versa. Further can use this data to track a specific bird should the species be sufficiently rare.

User 3 – Admin: Manually confirms Conservationists authenticity and will interact with the administrative systems.

Constraints:

1. Only a mobile application will be developed.
2. Only South African bird data through SABAP2 will be available.
3. The application will only be available in English.
4. Application requires internet connection at least once to download bird information for offline use.

Functional Requirements:

An Asterix(*) indicates an optional requirement.

1. User Profile Creation and Management:

1.1. Register an account using either their email or Google account.
1.2. Login into their account using social sign-in (Google account) or their normal details.
1.3. Manage their profile.

2. Interactive Map:

2.1. Display an Interactive South African map.
2.2. Show user's current location.
2.3. View bird species by pentad.
2.4. View the percentage chance that the user has of seeing a specific bird.
2.5. View likely bird species near the user.
2.6. View a heatmap of the birds locations.

3. Identify and Find Birds:

3.1. Search for a specific birds location data.
3.2. View Pentad information of bird sightings.
3.3. Take a picture of a bird to identify it*.
3.4. Record the sound a bird makes to identify it*.

4. Enrich User Knowledge:

4.1. Play a bird quiz minigame*.
4.2. View all text-based information regarding a birds location.
4.3. See what a bird looks like*.
4.4. Hear what a bird sounds like*.

5. Allow Users to Compete:

5.1. Earn experience points based on the rarity of the bird sighted to increase their level.
5.2. Earn special badges based on challenges.
5.3. Participate in daily, weekly, and monthly challenges.
5.4. Report users*.

6. Specialised Data View for Conservationists:

6.1. View the bird sighting data analytically.
6.2. Make note of the existence of a nest or hatchlings.
6.3. View population projections.
6.4. Filter by birds that have a tag.
6.5. View unexpected sightings. (rare bird or bird in wrong area)
6.6. Track specific areas for species of birds.

7. Manage and Process Data from Database:

7.1. Contribute to the user bird sighting database.
7.2. Ingress of data from SABAP2.
7.3. Processing of data from SABAP2.
7.4. Verify user sighting.

8. Administrative System and Users:

8.1. Authorise a user as a conservationist.
8.2. Authorise new admins.
8.3. Manage database systems.
8.4. Ban users*.

Subsystems:

1. User Profile Creation and Management
2. Interactive Map
3. Identify Birds
4. Enrich User Knowledge
5. Allow Users to Compete
6. Specialised Data View for Conservationists
7. Manage and Process Data from Database
8. Administrative System and Users

Non-Functional Requirements:

1. Performance

1.1. The system should be able to handle concurrent users.
1.2. The system should be able to handle concurrent messages and notifications.
1.3. The system should be able to handle low network speeds.
1.4. There should be no more than a two second response time.
1.5. Notifications should be sent in real-time.

2. Reliability

2.1. The system should be available 20/7 (Four-hour maintenance window is acceptable between 22:00 and 02:00).
2.2. Should be able to recover from crashes and failures.
2.3. The system must survive upgrades and device migration.
2.4. The system should be scalable past 10 000 active users per month*.
2.5. The system must function with low bandwidth.

3. Security

3.1. Be able to protect user data from unauthorised access.
3.2. Be able to protect user data from unauthorised modification.
3.3. Be able to protect user data from unauthorised deletion.
3.4. Be able to protect user data from unauthorised disclosure.

4. Maintainability

4.1. The system should be easy to debug.
4.2. Be able to be updated without affecting the user experience.
4.3. The system should be easy to maintain.

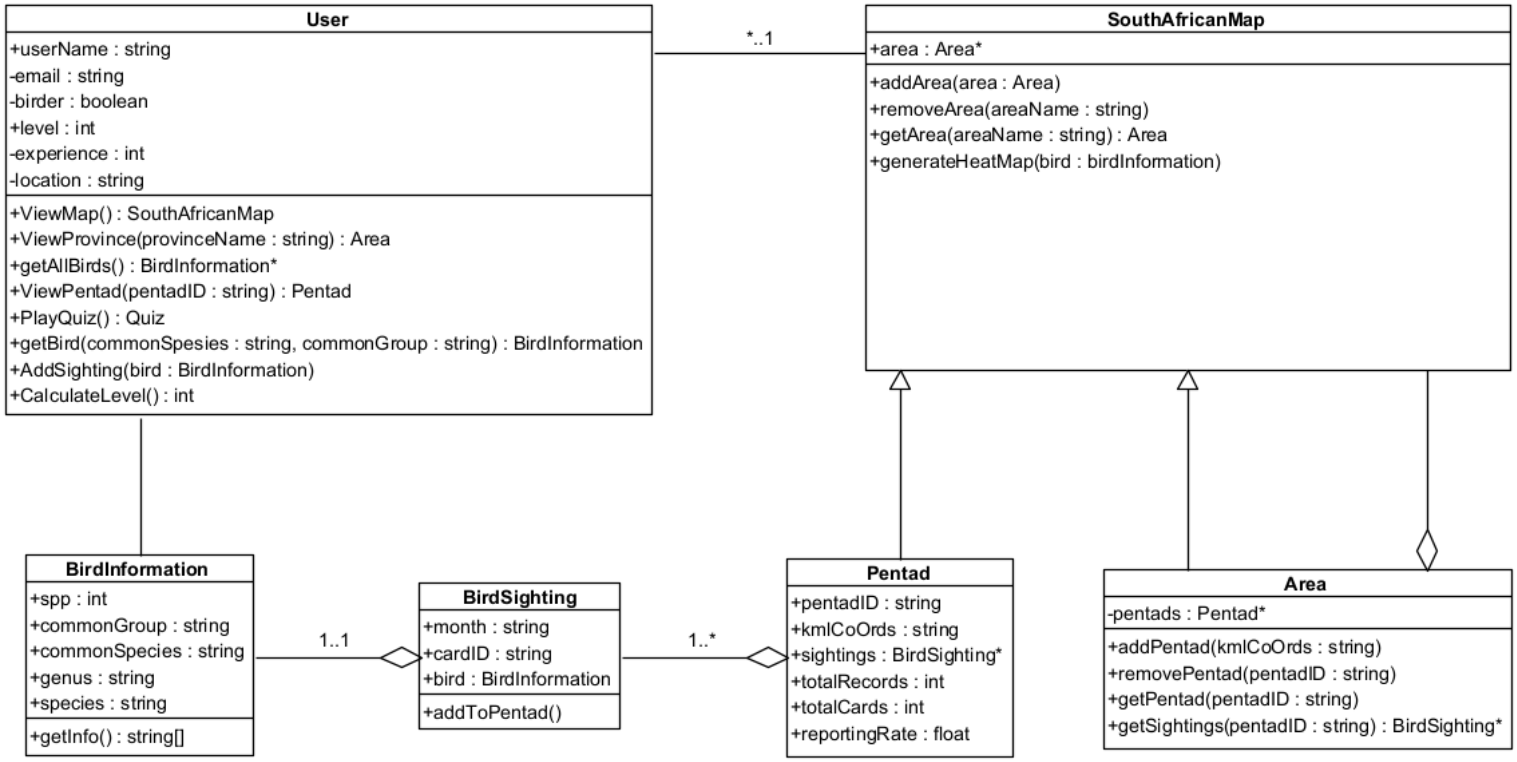
5. Usability

5.1. The system should be easy to use.
5.2. The system should be easy to navigate.
5.3. The system should be easy to understand.
5.4. The system should be easy to learn.

6. Efficiency

6.1. The system should not drain more than 5% battery per hour.

UML Class Diagram:

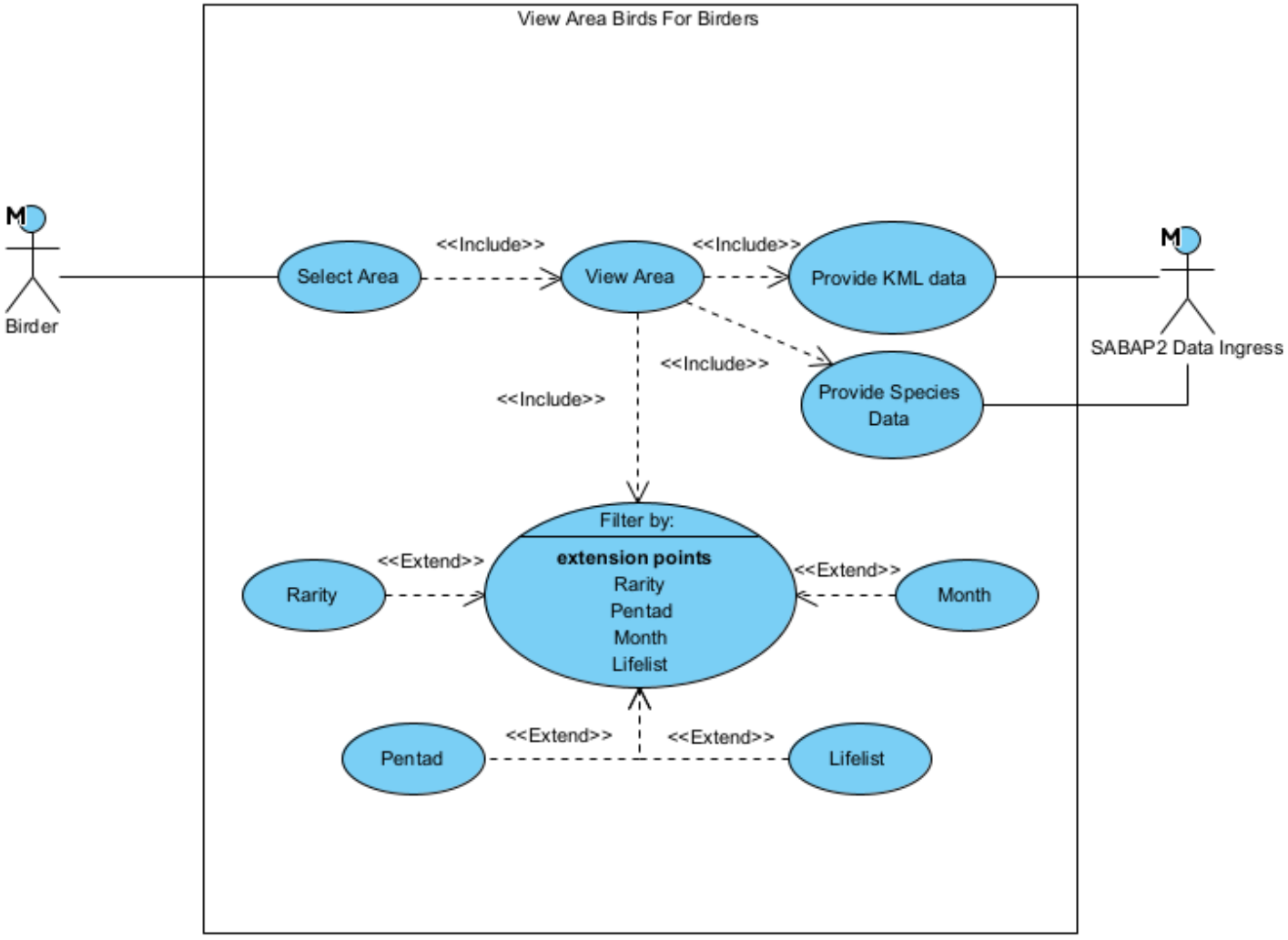


Use Cases:

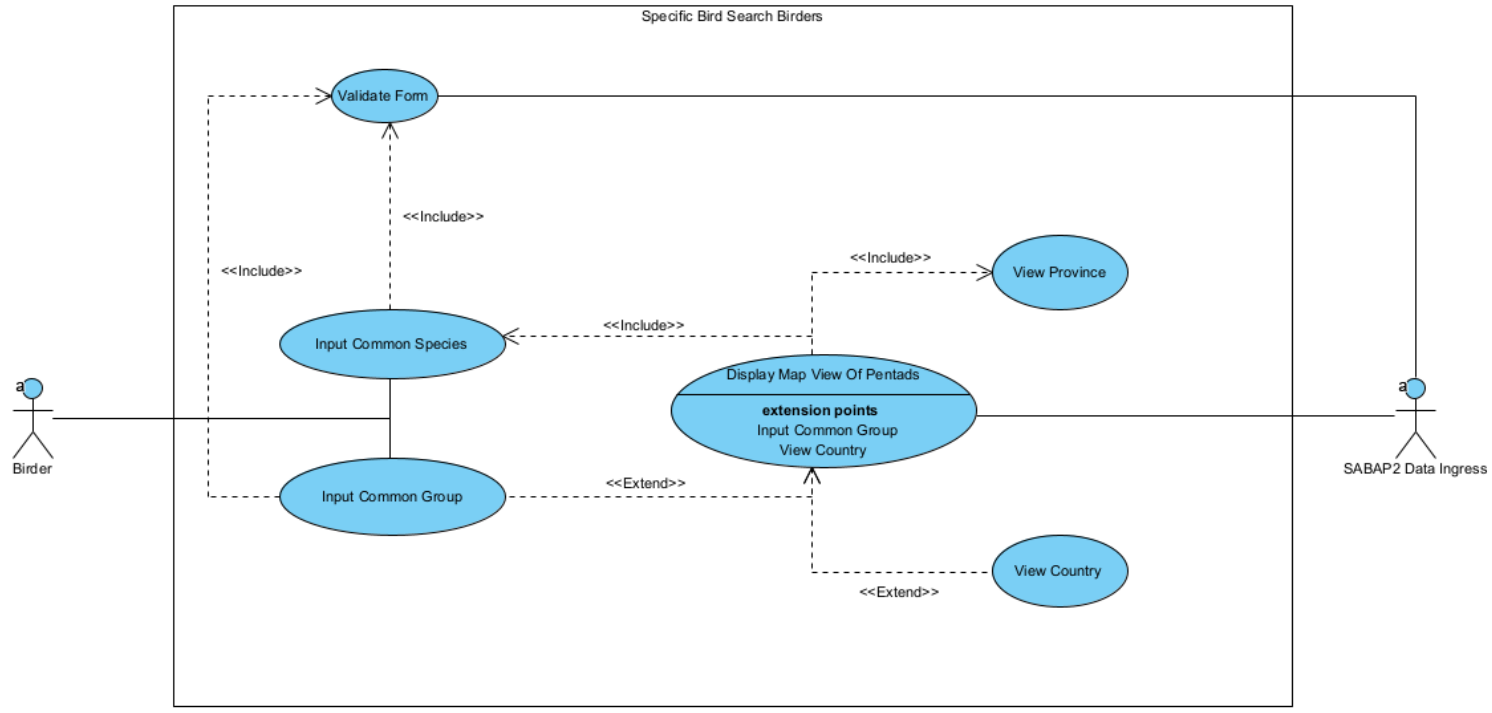
1. View Province birds. (Birder)
2. Search for a specific bird. (Birder)
3. User Login. (Conservationist and Birder)

Use Case Diagrams:

View Province Birds for Birders:



Specific Bird Search for Birders:



User Login: