

Stone Identification App

High-Level Architectural Design

Genevieve Okon (okong), Sydney Lieng(liengsn),
Niko Savas(savasn), Nick Lago(lagond),
Eric Le Fort(leforte)

March 6, 2016

Contents

1	Introduction	2
1.1	Purpose	2
1.2	System Description	2
1.3	Overview	2
2	Use Case Diagram	3
2.1	Session	3
2.2	Operation	3
2.3	Add Search Criteria	3
2.4	Refine Search Results	3
2.5	View Rock Information	3
2.6	View Collection	3
2.7	Save Rock	3
3	Analysis Class Diagram	4
4	Architectural Design	4
4.1	System Architecture	4
4.2	Subsystems	5
5	Class Responsibility Collaboration (CRC) Cards	6
A	Division of Labour	8

1 Introduction

This document will explain the design of the Stone Identification App using, Use Case Diagram, Analysis Class Diagram, Architectural Design and Class Responsibility Collaboration Cards.

1.1 Purpose

The purpose of this document is to describe the design architecture of the Stone Identification App. This will be achieved through a series of different diagrams and analysis techniques. The intended audience for this document is the software implementation team as well as the client.

1.2 System Description

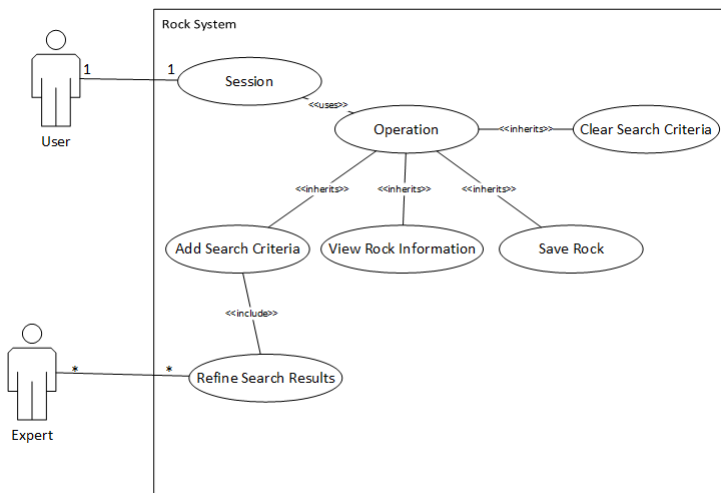
The system that will be discussed in this report is that of the Stone Identification App. The system will be composed of multiple expert classes. Each of these classes will have access to a database containing a list of rocks and the details pertaining to those rocks. When a customer inputs information about a rock the system will use these experts to give a list of possible rocks they may have found. The user will be able to keep information on rocks they have found on a separate database held on local storage devices. The system will be designed as an android application. It will be able to be used on any android devices and will be purchased through the android app store.

1.3 Overview

This document has 4 sections not including this one. Each section either contains design diagrams or an explanation that will further describe the architecture of the Stone Identification App. Each section will provide insight into a key aspect of design with the goal of preparing the software team to implement the design.

- **Use Case:** This document will show who interacts with the system as well as the intended results of those interactions through Use Case Diagrams
- **Analysis Class Diagram:** The Analysis Class Diagram will outline which classes will be a boundary, controller or entity class and show their connections to one another
- **Architectural Design Class:** This section will show the overview of the overall design, and justification for the choice of this design
- **Class Responsibility Collaboration Cards** The use of these cards will show what the classes that make up the Stone Identification App key responsibilities are and what other classes they interact with

2 Use Case Diagram



2.1 Session

When the user opens the app, a session is started. During the session, the user can manage their search criteria to identify a rock, with information such as colour, size, and texture. The session is terminated when the user closes the app or is finished with their search.

2.2 Operation

The operation is an abstract use case. It is extended by clear search criteria, add search criteria, view rock information, view collection, and save rock.

2.3 Add Search Criteria

The user can add information about the rock to the search criteria to refine the list of possible matches. The list of possible matches is generated by both internal and external experts ("internal experts" are built in to the application, while "external experts" are actors upon the application providing context and data).

2.4 Refine Search Results

This use case is implemented by all external experts contributing to the application. Given a set of inputs, an expert can refine a set of search results to only include those that match the inputs.

2.5 View Rock Information

The user can view detailed information about any given rock included in the list of matches or their collection. The information displayed is sourced from the data store of the application.

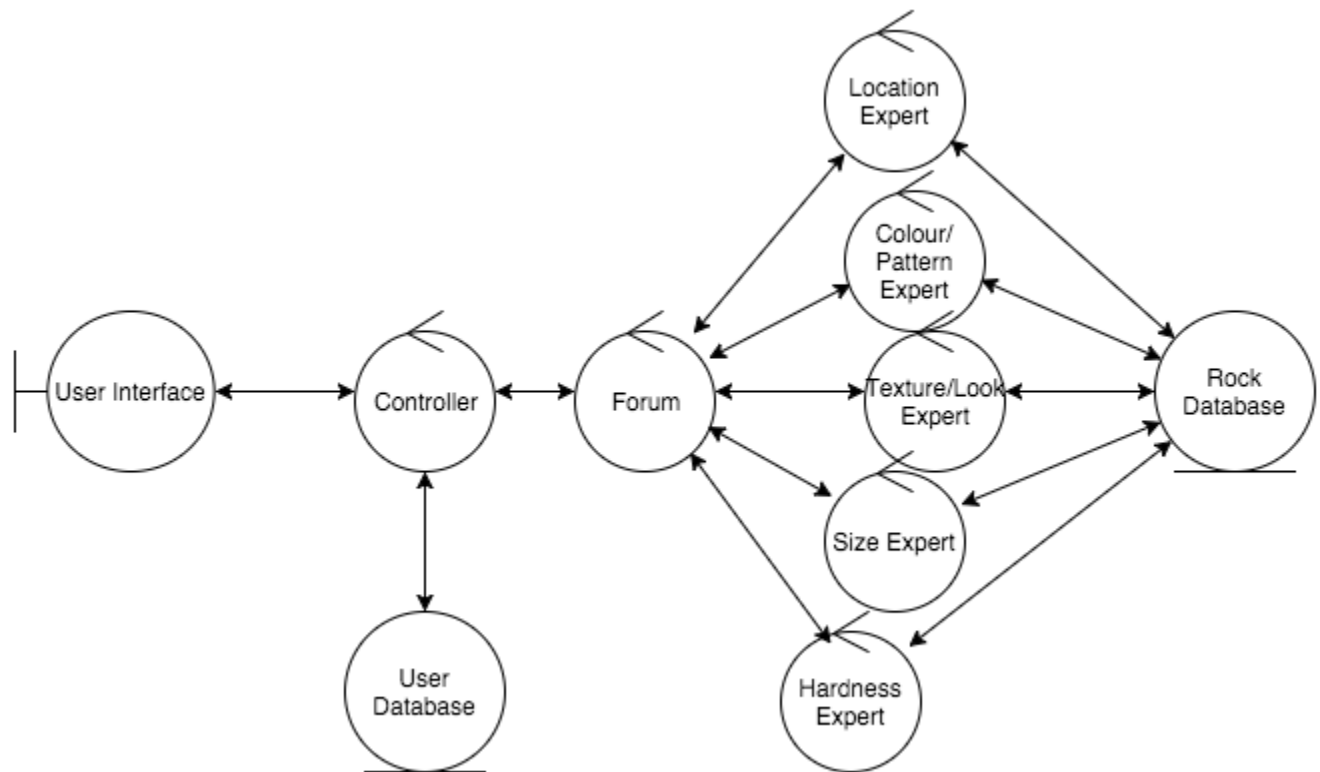
2.6 View Collection

The user can view their collection of saved rocks. This provides a shortlist of rocks that have been added using the "Save Rock" operation.

2.7 Save Rock

The user can, at any time, add a rock to their collection.

3 Analysis Class Diagram



4 Architectural Design

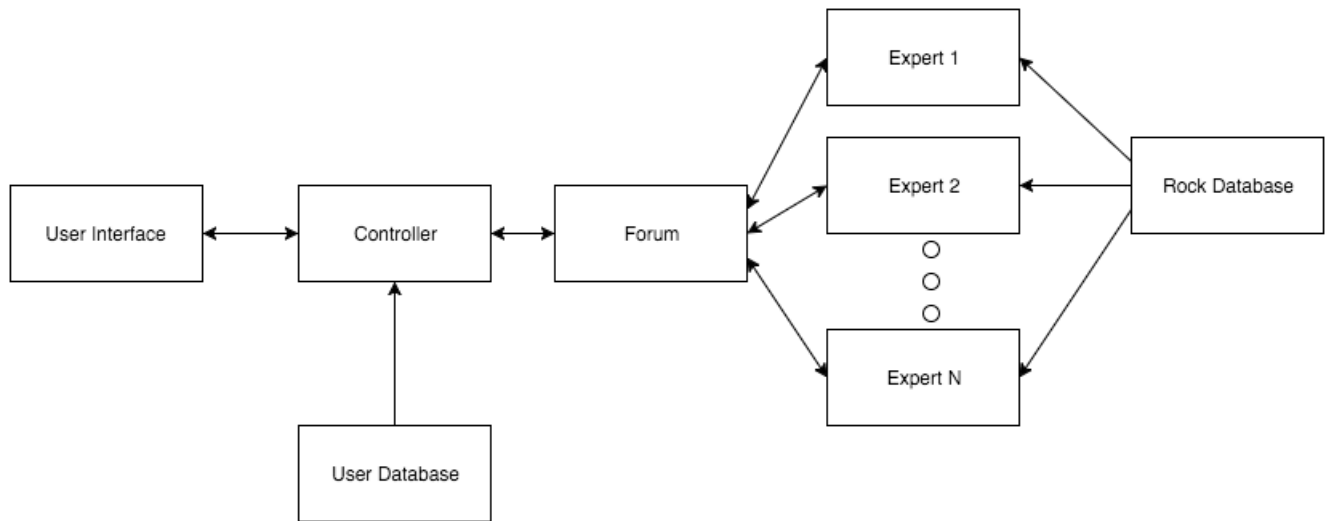
This system will implement a blackboard architecture

4.1 System Architecture

There will be a series of experts that will post the results of their findings to the forum. This forum will be used to consolidate the results. The best rock or rocks will be picked according to how many experts say they are a potential match and then showed to the user.

This blackboard architecture model was originally designed as a way to handle complex, ill-defined problems, where the solution is obtained using the sum of its parts. This model is the best choice for this task since it is designed to handle multiple sources of data and determine its results based off those streams.

The following diagram illustrates the proposed architecture for this system:



4.2 Subsystems

Expert - The experts will take in a criteria such as colour, texture or hardness and return a list of rocks that match that criteria.

Forum - The forum will receive the lists generated by each of the experts and retain those that are common to all lists.

Interface - The interface will be used to provide information to and receive commands from the user.

Rock Database - The database will be used to store essential data as well as providing tools to access that data.

User Information Database - The database will be used to store information pertaining to the user as well as providing tools to access that data.

Control - The control will grab data from the forum, clear data on the forum and alter what the forum is asking for.

5 Class Responsibility Collaboration (CRC) Cards

This section should contain all of your CRC cards.

Class Name: Controller	
Responsibility:	Collaborators:
Handle the information requests	Forum
Handle information outputted	Forum
Handle recent rock searches	User Database
Handle initial registration request	User Database
Handle login/logout requests	User Database
Send search requests to Forum	

Class Name: User Interface	
Responsibility:	Collaborators:
Allow user to login/logout	Controller, User Database
Allow user to enter information for different rock qualities	
Allow user to request to view rock history	Controller
Send information about rock search to Controller	
Show visual of all requested information	Controller
Receive and parse information from Controller	

Class Name: User Database	
Responsibility:	Collaborators:
Update password	Controller
Keep user's password encrypted and secure	
Send success/failure in login to Controller	
Handle all found rock history requests	
Send/receive requests to/from Controller	

Class Name: Forum	
Responsibility:	Collaborators:
Know what experts are needed for the search	Controller
Know all experts distinctions (LocationExpert, ColourExpert, TextureExpert, SizeExperts, HardnessExpert)	
Send/receive information to/from Controller	
Using tables gathered from each expert, create a master list of rocks without duplicates	

Class Name: LocationExpert	
Responsibility:	Collaborators:
Use Location information given from Forum to query RockDatabase	RockDatabase
Send information gathered back to Forum	

Class Name: ColourExpert	
Responsibility:	Collaborators:
Use Colour/Pattern information given from Forum to query RockDatabase	RockDatabase
Send information gathered back to Forum	

Class Name: TextureExpert	
Responsibility:	Collaborators:
Use Texture/Look information given from Forum to query RockDatabase	RockDatabase
Send information gathered back to Forum	

Class Name:SizeExpert	
Responsibility:	Collaborators:
Use Size information given from Forum to query RockDatabase	RockDatabase
Send information gathered back to Forum	

Class Name: HardnessExpert	
Responsibility:	Collaborators:
Use Hardness information given from Forum to query RockDatabase	RockDatabase
Send information gathered back to Forum	

Class Name: RockDatabase	
Responsibility:	Collaborators:
Receive requests from experts and query through the database for matches	LocationExpert, ColourExpert, TextureExpert, SizeExperts, HardnessExpert
Send back corresponding rows to the expert classes	

A Division of Labour

Team Member	Contribution	Signature
Nick	Introduction	
Niko	Use-Case Diagrams	
Genevieve	Analysis Class Diagram	
Eric	Architectural Design, creation of final document	
Sydney	CRC Cards	