Deliverable #2

SE 3A04: Software Design II – Large System Design

1 Introduction

This section of the High-Level Architectural Design provides an overview of everything included in the High-Level Architectural Design.

1.1 Purpose

The purpose of this document is to provide a representation of the structure of the data and program components that will be required to build the Movie Matcher software. This document will primarily be utilized by the developers of the software application and will be reviewed by the teaching assistants.

1.2 System Description

Movie Matcher is a mobile software application that will provide a platform through which users can find the name of a movie after being given specific details the user knows about the movie. The application must also display information relevant to any movies matching the users criteria. If the user successfully finds the movie that they were searching for, they should be directed to all nearly theaters that are currently screening the movie. If the movie is no longer in theaters, the user should be provided with links to purchase the movie. The primary goal of this software is to help users identify movies based on movie attributes the user inputs into the system. The software cannot guarantee to always find the correct result, or any result whatsoever if none of the inputted attributes match the movies in the database. The software will attempt to locate nearby movie theaters based on its interaction with the pre-implemented GPS in the users mobile device. The software will also require a stable internet connection to function and perform searches.

1.3 Overview

This document is divided into 5 separate sections. The first section is this introduction. The second section consists of the Use Case Diagram, which is a representation of a users interaction with the system. The Use Case Diagram will be accompanied with a text description. The third section contains the Analysis Class Diagram which is a static structure diagram that serves the purpose of describing the structure of the software, by showing the systems subclasses, methods, attributes and the relationships between objects. The fourth section provides an overview of the overall architectural design of the software application. It will show the division of the system into subsystems with high cohesion and low coupling, thereby following good software engineering principles. The fifth and final section will display the Class Responsibility Collaboration (CRC) Cards. A CRC card will be provided for each identified class. By organizing the document into 5 distinct sections with clear diagrams for each section, the audience will find the document easy to read and understand
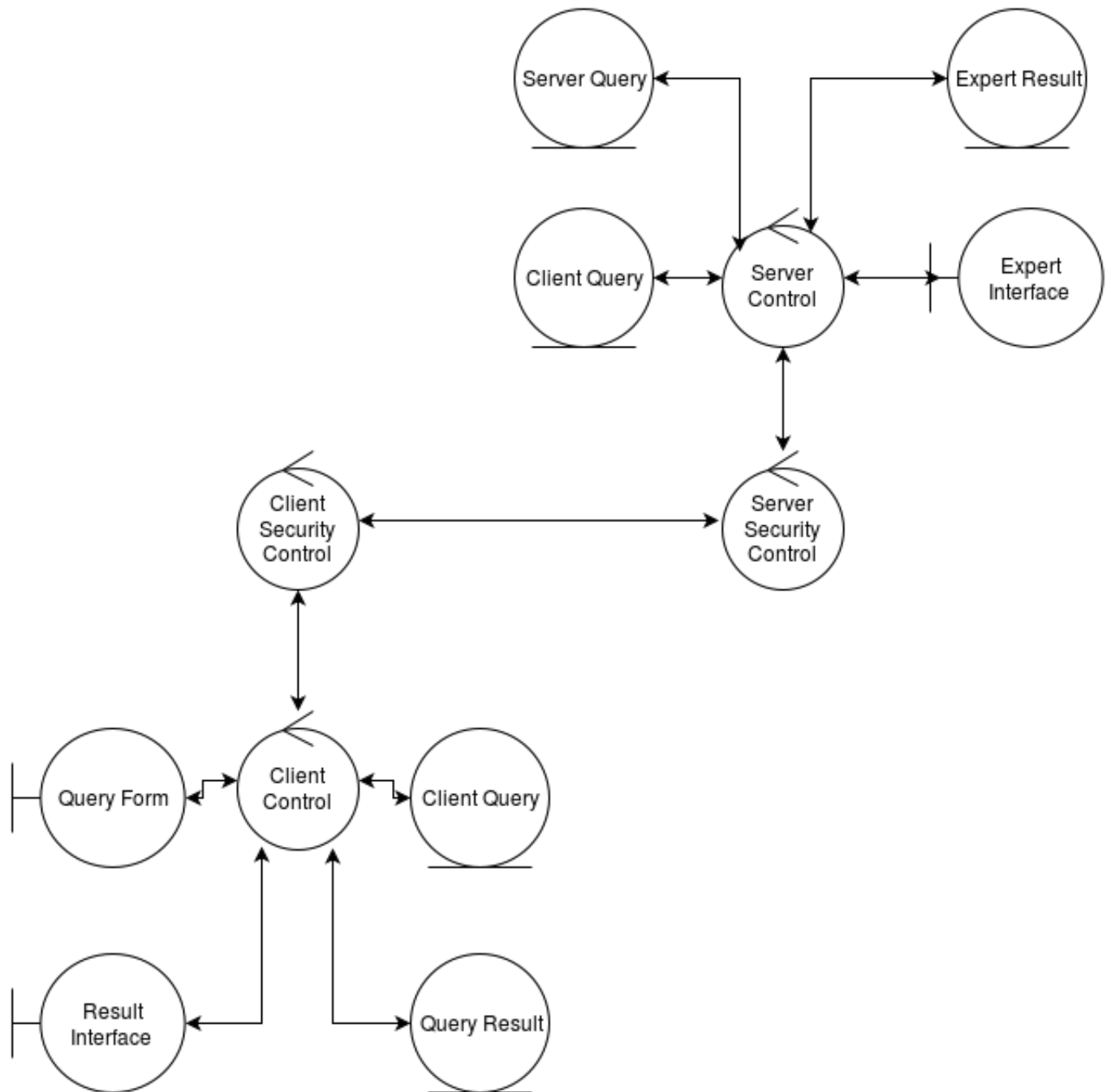
2 Use Case Diagram

a) The use case for the client side of the application.
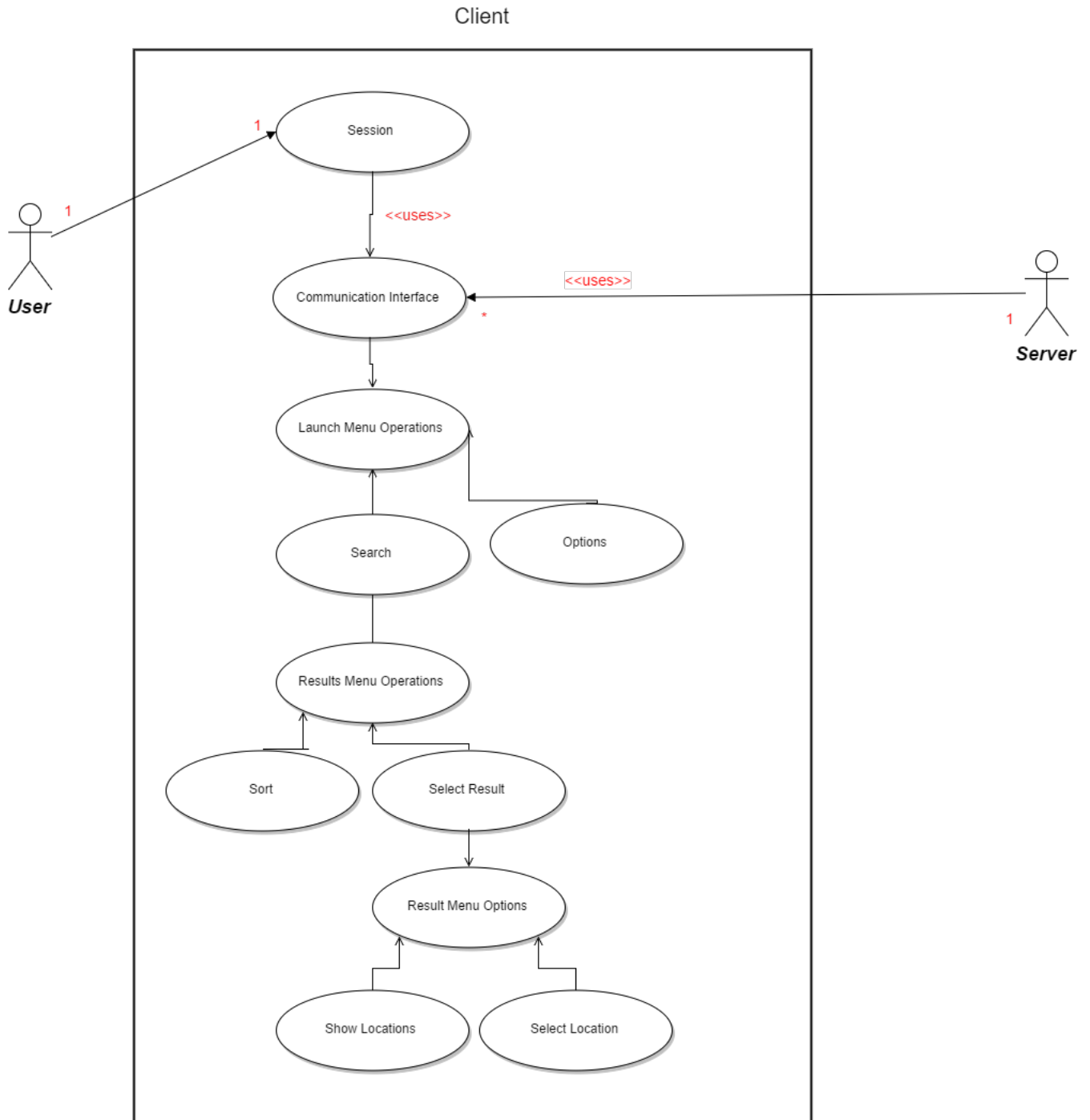
Client

b) The use case for the server side of the application.
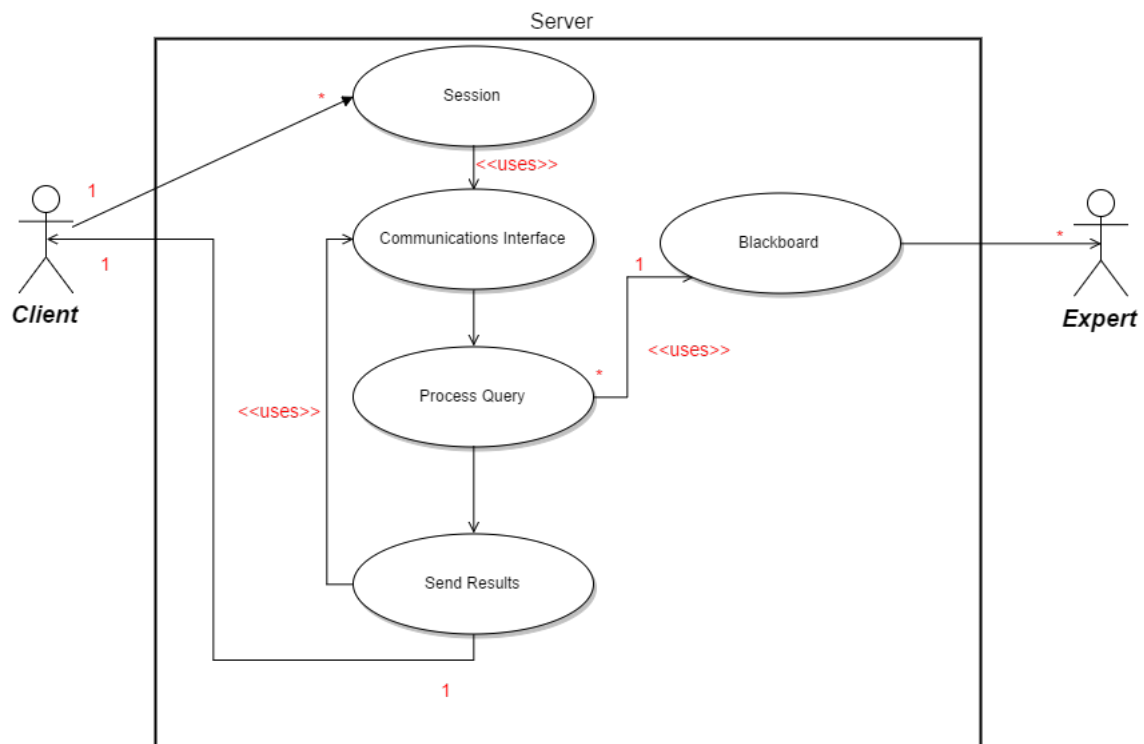
3 Analysis Class Diagram

4 Architectural Design

4.1 System Architecture

The overall architecture of the system is a hybrid between a Client-Server Architecture and a Blackboard Architecture. Both of these architectures are needed to make the design more robust and simple.

In a Client-Server Architecture, there are many clients that communicate with a server. In the case of this system, this architecture makes sense because there is a client with a GUI that the users can use to

make searches and have information displayed to them. There is also a server component that will contain large databases of information which would be too large to store in the client. The server can respond to a client's query by using the database of information. The reason why a Client-Server architecture is necessary is that the databases of movie information are very large and it would be too much to download considering that this system is a mobile application. This architecture allows the data to be stored on the server and only relevant parts are communicated to the client.

In the Blackboard Architecture there is a blackboard subsystem that stores data, and there is a knowledge source subsystem that stores domain specific data. In this system, the blackboard and knowledge subsystems are part of the server system. The blackboard is on the server and will contain the information about a movie, and what movie each expert thinks corresponds to this data. The knowledge source will be databases stored on the server because databases of movies are very large, much too large to store in the client system. The knowledge subsystem is the experts. The reason for using a Blackboard Architecture is that it allows for more experts to be easily added. It also makes it easy to combine the results from each individual expert in a centralized location.

4.2   Subsystems

Blackboard Provides information to the experts and combines the results.

Communication Handles communication between client and server. Includes necessary encryption.

Knowledge Contains many experts, each has a specific method of going from information about a movie, to what movie that information is about. Gets the information from the Blackboard subsystem and returns the result to it as well.

UI Provides the user with a graphical user interface so they can interact with the client.

5   Class Responsibility Collaboration (CRC) Cards

This section should contain all of your CRC cards.

a) Provide a CRC Card for each identified class

b) Please use the format outlined in tutorial, i.e.,

Class Name: Client Query	
Responsibilities:	**Collaborators:**
Decrypt the encrypted client query information and send it to server control.	Server Control.

Class Name: Server Query	
Responsibilities:	**Collaborators:**
Format the decrypted client query to be used to query the experts.	Server Control.

Class Name: Client Results	
Responsibilities:	**Collaborators:**
Retrieve the expert results and stores it.	Server Control.

Class Name: Client Interface	
Responsibilities:	**Collaborators:**
Retrieve the server query. Use the server query to search the experts for information regarding the search query. Send the information retrieved from the expert search to the server control	Server Control.

Class Name: Server Control	
Responsibilities:	**Collaborators:**
Handle client query, server query and expert results being sent and retrieved from other classes.	Expert Interface, Expert Results, Server Query, Client Query.

Class Name: Server Security Control	
Responsibilities:	**Collaborators:**
Handle client query and query result being sent and retrieved from other classes.	Server Control, Client Security Control.

Class Name: Client Security Control	
Responsibilities:	**Collaborators:**
Handle client query and query result being sent and retrieved from other classes.	Server Security Control, Client Control.

Class Name: Query Form	
Responsibilities:	**Collaborators:**
Handle click events of selecting a search criterion. Handle click events of the search button. Send the query form information to client control.	Client Control.

Class Name: Client Query	
Responsibilities:	**Collaborators:**
Encrypt the client query information and sends it to client control.	Client Control.

Class Name: Client Control	
Responsibilities:	**Collaborators:**
Handle client query information and client query results being passed from one class to another.	Client Security Control.

Class Name: Result Interface	
Responsibilities:	**Collaborators:**
Retrieve the client query results. Display the client query results to the user.	Client Control.

Class Name: Query Result	
Responsibilities:	**Collaborators:**
Decrypt the encrypted client query results and sends it to client control.	Client Control.

A   Division of Labour

Contributions			
Name	**Student Number**	**Contribution**	**Signature**
Joshua			
Keyur			
Justin			
Bilal			
Shaad			
Abdullah			