

Design Document

Conference Management System project

SuperCheapSoftware Group

Contents list:

(1) Object Classes design:

1.1 General Description

1.2 User Interface

1.3 UML diagram for the classes

(2) Database design:

2.1 General description

2.2 Database contents

2.3 High level Data Model for database

2.4 Conference system Database architecture

2.5 Database Model entities

2.6 Database architecture diagram

(3) Graphical User Interface

3.1 General Description

3.2 Examples

(1) Object Classes design:

1.1 General Description :

The conference management system will be built of several classes and objects. These classes are Conference class, Paper class, Review class and A User interface that will be implemented by Program Chair class, Area Chair class, Author class and Reviewer class.

1.2 User Interface :

The user interface will be implemented by four classes which are Program Chair class, Area Chair class, Author class and Reviewer class.

```
import java.util.*;

public interface User {

    String userID;

    String emailAddress;

    String password;

    String name;

    /**
     * method for the user to login to the conference on
     * an existing account
     * @param email the users registered email address
     * @param password the users chosen password
     */
}
```

```
public void Login(String email, String password) ;

/**
 * method for user to logout of the conference tool
 * @param userID unique id number of the user to
 * logout
 */
public void Logout(String userID) ;

/**
 * method to search for papers in the conference
 * @param keywords array of strings describing
 * topics in the conference
 * @return array of papers with matching keywords
 */
public List<Paper> Search(String[] keywords) ;
```

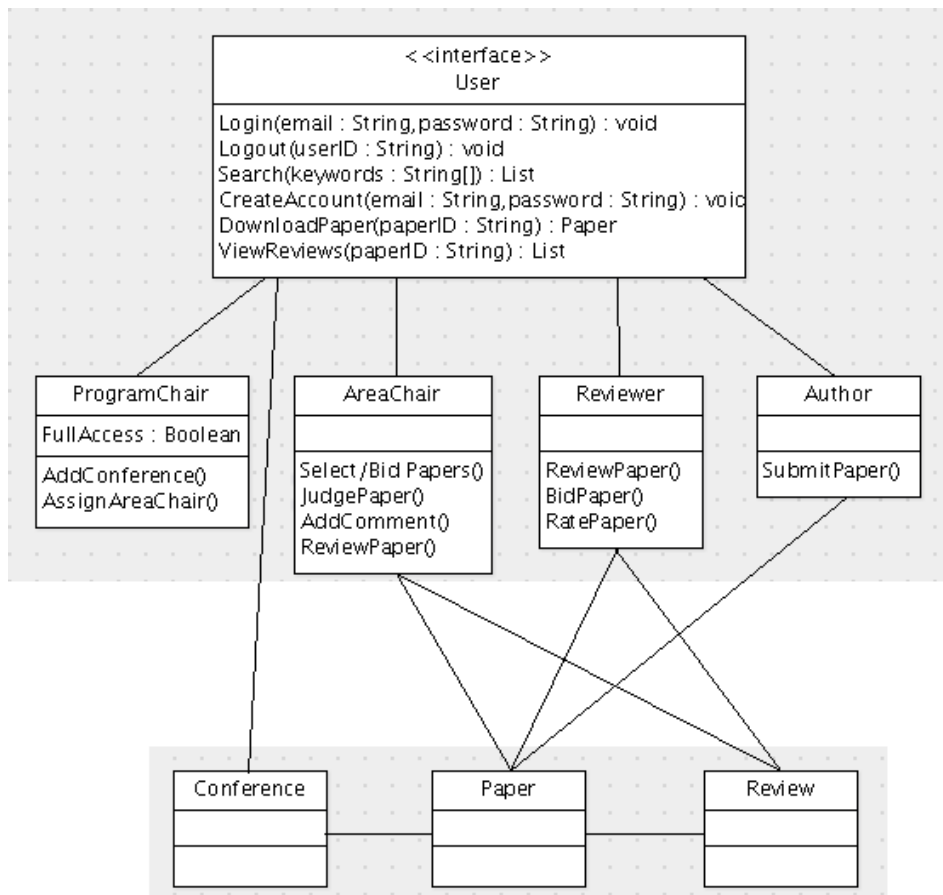
```
/**
 * method for the user to create an account for the
 * conference
 * @param email the users email address they use to
 * sign in
 * @param password users chosen password to login
 * with
 */
public void CreateAccount(String email, String
                               password);

/**
 * method for user to get a submitted paper from the
 * conference
 * @param paperID id number unique to the paper
 * @return the paper object
 */
public Paper DownloadPaper(String paperID);

/**
 * method to access reviews from a paper on the
 * conference
 * @param paperID id number of the paper wanting to
 * view
 * @return list of reviews submitted for that paper
 */
public List<Review> ViewReviews(String paperID);
```

```
}
```

1.3 UML diagram for the classes :



UML diagram

The UML diagram shows roughly the relationships between the classes. Also it shows some of the functions that will be implemented in some of the classes.

(2) Database design:

2.1 General description :

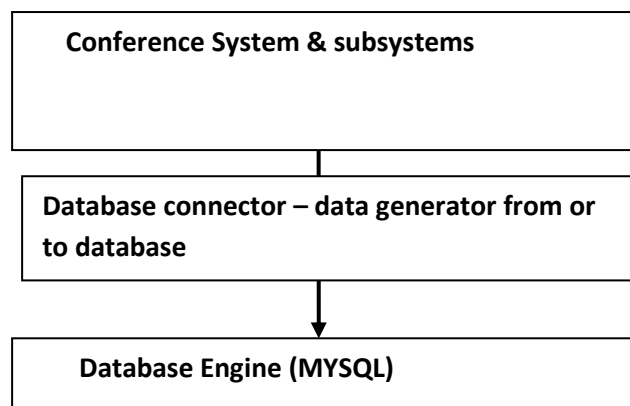
The following documents outline the entities, the database engine will use, and main key attributes of each entity. Then they outline the architecture of the database model and how it works to generate data from the MYSQL machine or to save data to it as well.

2.2 Database contents:

Our database is supposed to include:

- The conference details.
- All users and any user account can be retrieved from the database.
- Submitted, searching, and downloading paper
- Paper Review.

2.3 High level Data Model for database:

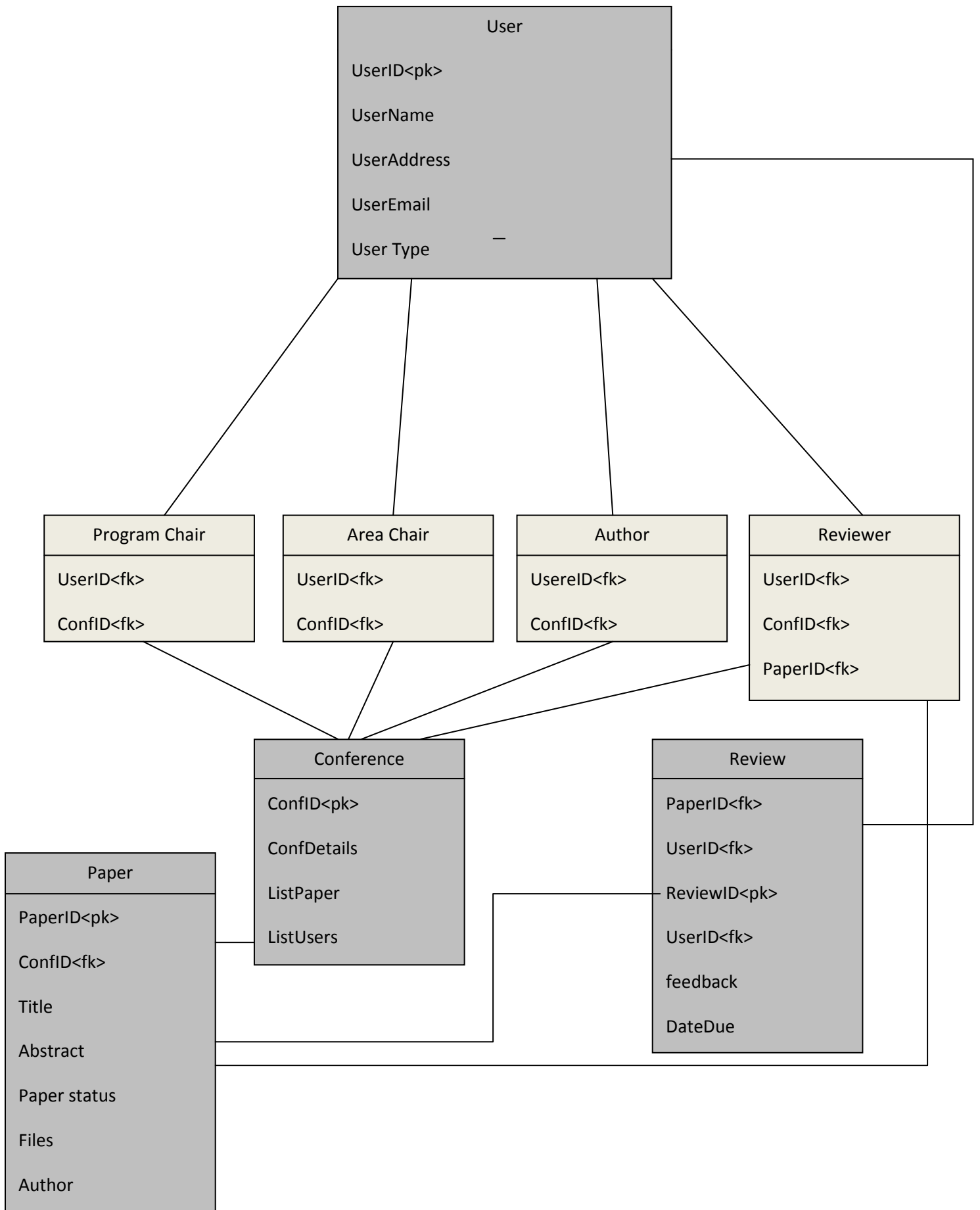


2.4 Conference system Database architecture:

This section illustrates the diagram in the next pages:

- The object oriented design systems and subsystems call the appropriate methods from current Database class when to add, get, delete or update papers, review or users.
- Database system uses connector to connect to MYSQL machine.
- SQL_Insertion _selection _engine class is to execute queries for inserting; selecting, updating and dropping any elements form MYSQL.
- Current Database class methods will call the methods from SQL_Insertion _selection _engine class to do the job.
 - Before calling methods, the Checking Function will check if the user or paper or review already exists i.e. it will check the unique elements like username or paper keywords are already existed.
 - The system then will not accept adding the new element to the MYSQL.
 - The system will ask the user to enter new unique elements that haven't been use before.

2.5 Data Model Diagram:

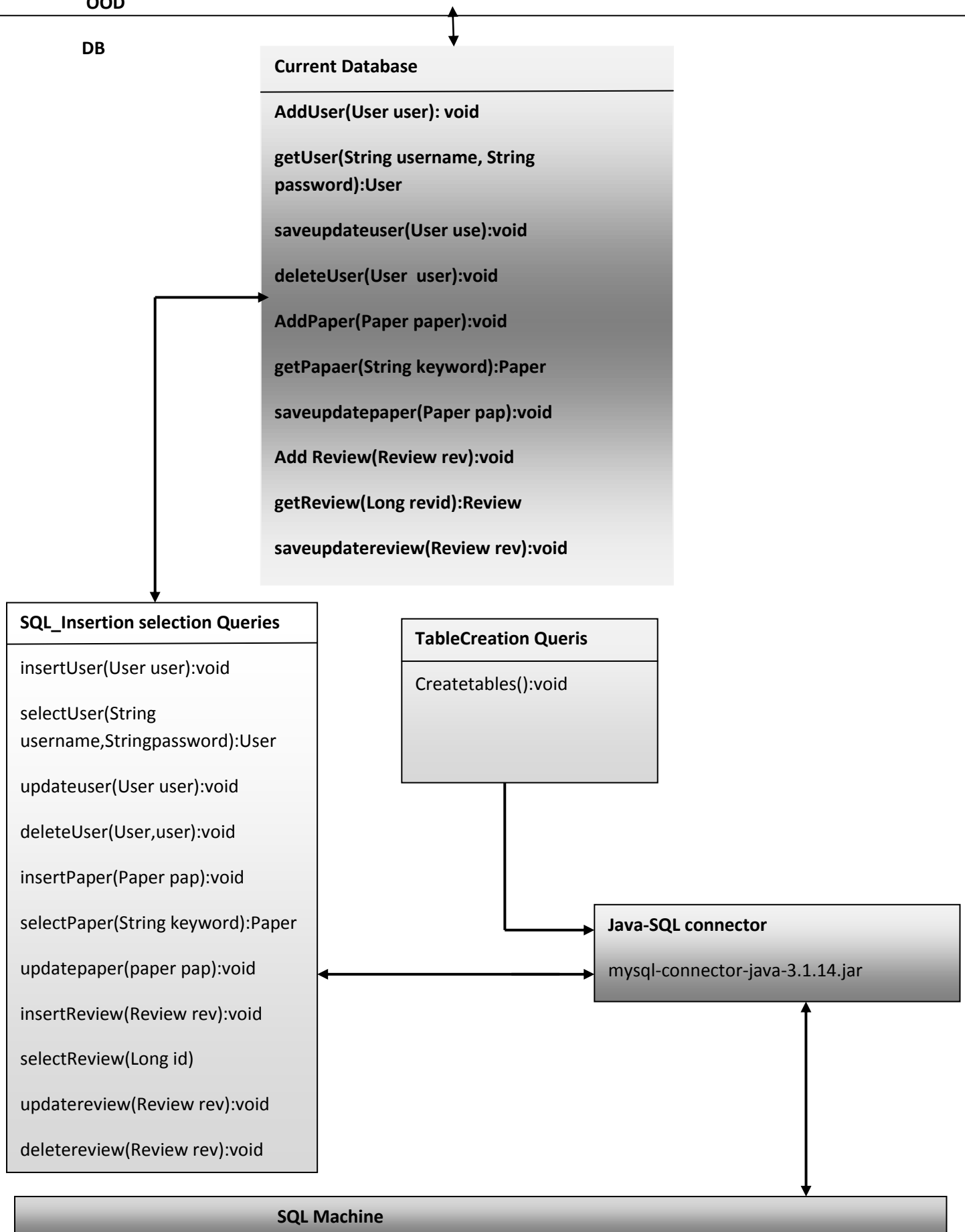


2.6 Database architecture diagram:

GUI

OOD

DB



(3) Graphical User Interface:

3.1 General Description:

The GUI will have several forms for ease of use. A limited selection of items through, for example; combo boxes rather than text fields will be used to avoid errors such as choosing the login type “whether an Author, Reviewer, Area Chair or Program Chair”.

Each of the different users will have a different interface to suit their part in the conference. Some users will be able to login as an author and also as a reviewer for example but not simultaneously.

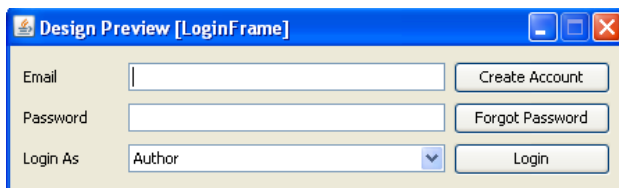
None of the forms hold the variables. Instead, a super class does which will also hold the methods to interact with the object oriented part.

3.2 Examples:

Below are examples of complete forms to date: (Login, the Create Account Wizard and Author Form)

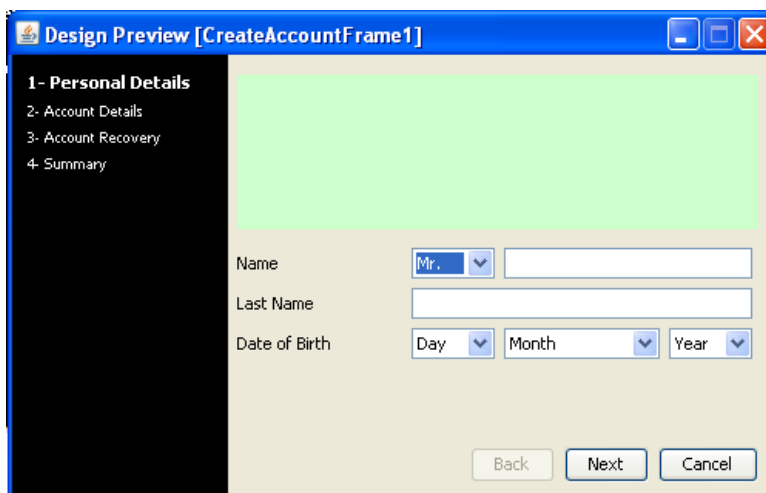
Note: many of forms below will be altered as we proceed and so this does not reflect the final quality of the GUI.

Login Forms:

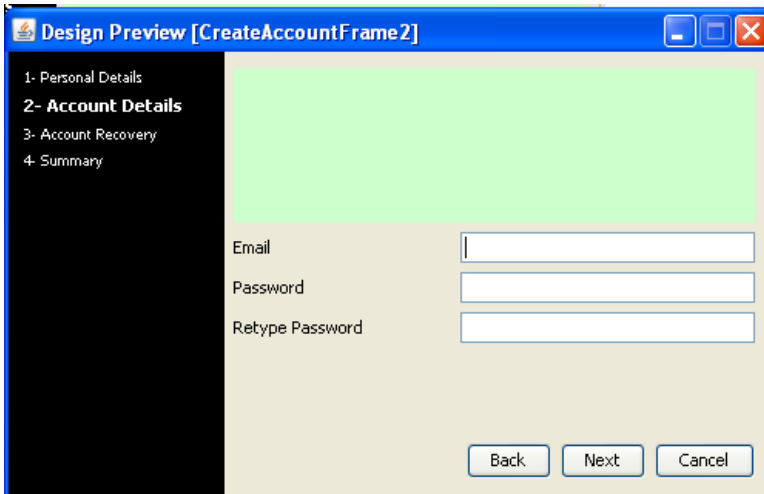


Above: Email and Password fields must be filled to proceed

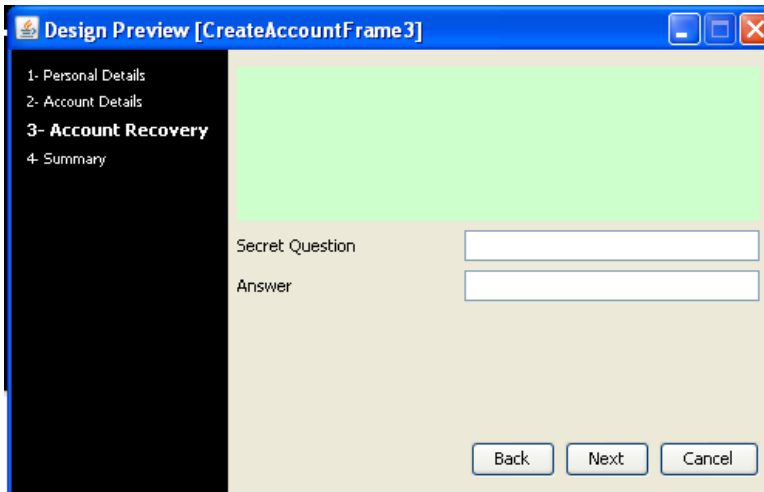
Create Account Wizard Forms:



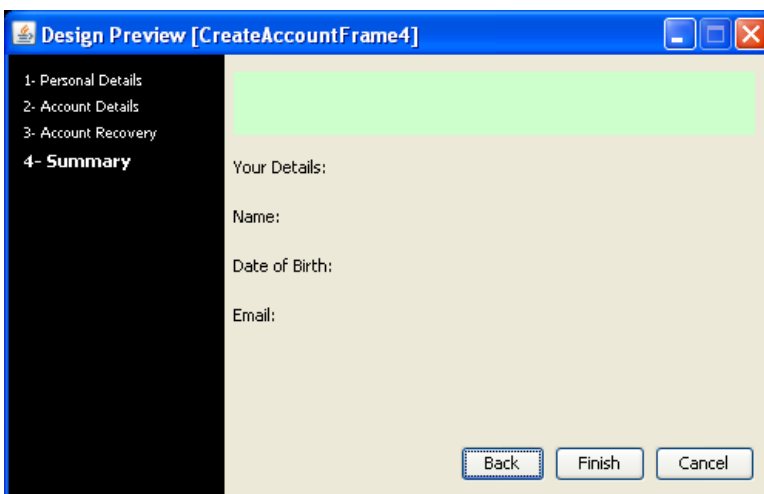
Above: all text fields and combo boxes must have values to proceed.



Above: password must be typed twice and must match to proceed.



Above: the secret question might be changed to a combo box rather than a text field... p.s. this whole feature might be removed in the future and other recovery options provided such as sending the password to the email address.



Above: final page to confirm the details before proceeding as they will be made permanent.

Author Forms:

The screenshot shows the 'Manage Papers' tab. It features two main sections for file management: 'Location of Paper to Upload' and 'Location to Download'. Each section includes a text input field, a 'Browse' button, and a corresponding 'Upload' or 'Download' button. At the bottom of the form, there are two buttons: 'Delete Highlighted Paper' and 'Submit Highlighted Paper'. To the right of the form is a table with two columns: 'Paper Name' and 'Status'. The table contains 13 empty rows.

Paper Name	Status

Above: the browse button will open a separate form from which the user can choose the file to upload or download.

The screenshot shows the 'Reviews' tab. It features three sub-tabs: 'Review 1', 'Review 2', and 'Review 3'. The 'Review 1' sub-tab is active and contains two input fields: 'Reviewer's ID:' and 'Feedback:'. To the right of the form is a table with two columns: 'Paper Name' and 'Status'. The table contains 13 empty rows.

Paper Name	Status

Above: the reviews tab (incomplete).