# Chapter Two: Design

## 2.1 Introduction

In this chapter I will decompose my application into smaller pieces, and design how to solve each piece.

## 2.2 Decomposition of the problem

Decomposition is the systematic breakdown of a complex problem or system into simpler parts. My solution is very detailed and contains many paths which can be displayed visually.

### 2.2.1 Decomposition Diagram

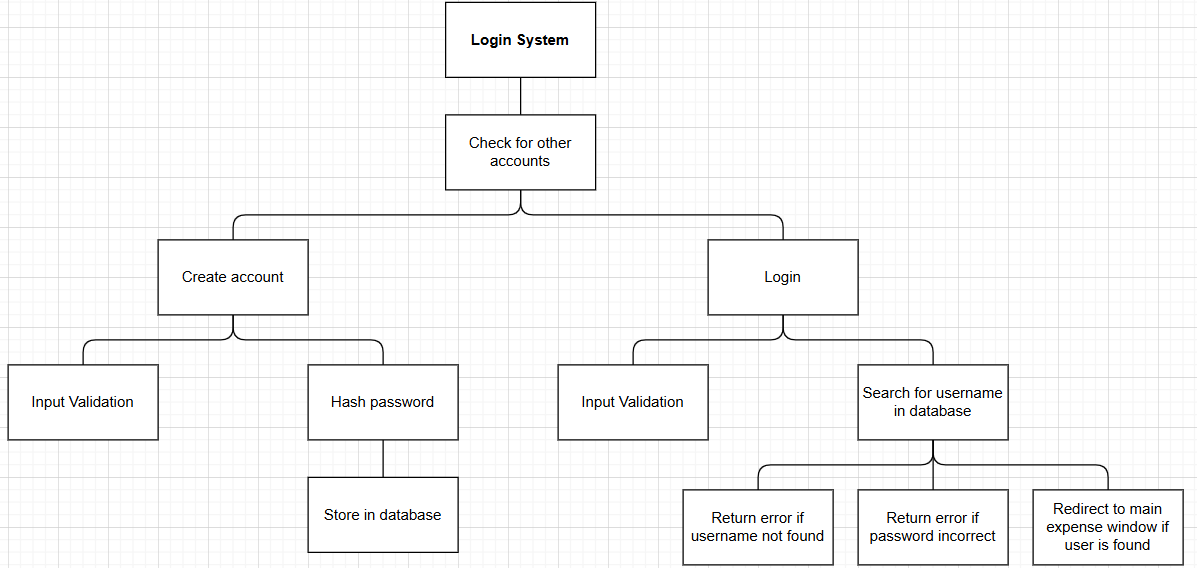
A diagram of a company

AI-generated content may be incorrect.

*Fig 2.2.1 Decomposition Diagram of Entire Solution*

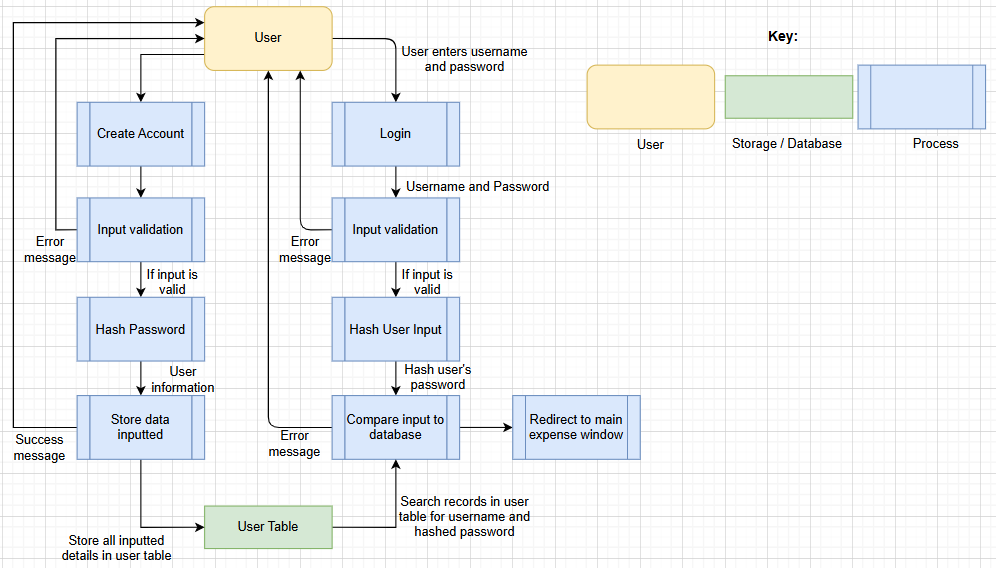
The problem has 3 main sections. The first is the login system where users can login using a user id and a password, or the user will need to create an account if no account has been made before. These accounts are used to access the main application. The second section is the settings, which should give helpful information to users who are unsure of what they are doing. There will also be options to change the app’s appearance and logout of the app. The third section is the main expense tracker, which allows users to access the account and expenses system, whilst giving options to create more accounts or extras such as database management, budgeting calculators and data importation / exportation.

### 2.2.2 Login System



*Fig 2.2.2 Decomposition Diagram for Login System*

### 2.2.2.1 Data Flow Diagram



*Fig 2.2.2.1 Data Flow Diagram for Login System*

### 2.2.2.2 Input Process Output Chart

|  |  |  |
| --- | --- | --- |
| **Subprocess** | **Description** | Justification |
| Create Account | When the application is loaded, a check of the database is completed and if there are no accounts detected, the user will have to make an account. Users need to input their first name, last name, email, phone number and password. The user id is provided to the user after this. | Each user needs to have a unique identification so the expenses they create, remove or edit are linked to their account. This means multiple people from the same company can use the application. Also, user levels can be used (default or admin) to give certain users more features. |
| Login | Entering a valid username and password is required to access the main application, so all users must login. |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### 2.2.3 Input Process Output

## 2.3 How All Solution Parts are Linked

### 2.3.1 State Diagram of the different forms/parts

### 2.3.2 How different functions /classes are connected

## 2.4 Database Design

### 

### Normalisation

### 

### Data Dictionary For Each Table

### Entity Relationship Diagram

### SQL Pseudocode

## 2.5 Design of other Parts of the Solution

2.5.1 Part ONE: ………………………..

### 

### Form/Screen Design and Layout

### Validation rules

### Key Variables/Data Structures /Classes

### Algorithms and PseudoCode

### Test Plan for PART ONE ….

2.5.2 Part TWO: ……………………

### Form Design and Layout

### Justification of Validation rules

### Key Variables/Data Structures /Classes

### Algorithms and PseudoCode

### Test Plan for PART TWO

2.5.3 Part THREE: …………………….

### Form Design and Layout

### Validation rules

### Key Variables/Data Structures /Classes

### Algorithms and PseudoCode

### Test Plan for PART THREE

2.5.4 Part FOUR: …………………………

### Form Design and Layout

### Validation rules

### Algorithms and PseudoCode

### Key Variables/Data Structures /Classes

### Test Plan for PART FOUR

## 2.6 Stakeholders involvement

## Testing plan to inform evaluation