UseCase 1.10 User Login

Alternate Flows: **AF1: Forgot Password**

The use case starts when the actor has not logged in and clicks Forgot Password link.

1. The System displays the Forgot Password screen.
2. The Actor enters his email address.
3. The Actor clicks on Send Password.
4. The use case ends when the System sends the new password to the actor's email address.

AF2: Change Password

The use case starts when the My Account action is invoked. The actor is already logged in.

1. The System displays the My Account screen.
2. The Actor clicks on Change Password.
3. The Actor enters the current password and the new password.
4. The Actor clicks on Change Password.

The use case ends when the System saves the new password.

Exception Flows: none

Special Requirements: None

There are many ways of creating use cases; we can also create them diagrammatically. But to keep things simple, we can follow the above use case, which is easier to understand. After the major use cases are covered, we can move to the next step in this elaboration phase, which is, prototyping.

Prototyping

The **Graphical User Interface** (**GUI**) of any project is one of the most critical areas in terms of its overall presence and credibility. And many projects are delayed because of repeated changes in the GUI throughout the project life cycle, adding to the frustration of the programmers. For web projects, designing a working prototype in HTML before starting to work on any other activity can be very helpful, for project stakeholders as well as developers. A working prototype means that the different HTML pages would be linked to each other (based on the use cases we covered earlier), and can use dummy data to give a realistic impression of the actual project.

A GUI prototype is not only a part of the **Proof of Concept** (**POC**), but also forms an important extension of the project requirements specifications, in a graphical sense. Properly-linked HTML pages with some dummy data showing the important business process flows can be an indispensable tool, aiding in the visual understanding of the project, as well as answering all kinds of technical and business-related questions. That is why it is highly recommended to develop a prototype before starting the actual coding for a project.

Class Model

The architect and the technical lead will create an object model of the system, highlighting all important entities and how they will interact. We will learn more about how to create an object model in the coming chapters of this book.

Database Model

A database model would be created based on the class model described above. This data model, along with the object model and use cases, will help provide the development team with clear instructions, and paths to the targets and objectives. It is very common for a data model to be created before a class model. It's also very common for these two steps to be completed at the same time, as they are very closely related.

Based on the use cases, the prototype and the object/data models, the project manager, along with the architect and the team lead, will develop a project plan for the construction phase, in which the coding takes place. This plan will highlight the milestones as well as list all of the important deliverables of that phase.

Project Construction: Programming and Development

When all of the primary documentation is complete, the actual development work starts. The technical team will study the use cases and the object and data models, and start planning the delivery cycles. Here, we can use one of the following famous development methodologies:

- SCRUM Development
- Waterfall Model
- Agile Development/Extreme Programming (XP)
- Iterative Development