Sports Center Membership System Configuration/Change Management Report

1. Purpose

This document describes the strategy of dealing with design and implementation issues, decisions, constraints, changes and other aspects of the system.

2. Introduction

Changes are inevitable during the development phase of a project. Changing circumstances, unexpected obstacles and requests can cause these changes. The most important part of this situation is how it affects the project.

The advantages and disadvantages of the changes should be compared to the project. This comparison should decide how the project will proceed.

It is important how the changes will affect time and cost. Some changes will have a positive impact, while others will have a negative impact. The project team should avoid these negative effects as much as possible.

To realize a change, it is necessary to know where the change originated. The steps to be followed later may vary. The sources of change can be classified as planned development, unexpected problems, or enhancements.

Ideally, a change is due to planned software development. When making these changes, you should also look at time and money effects. As long as it is not demanded, even if it is beneficial, all the changes will consume resources and disrupt the flow.

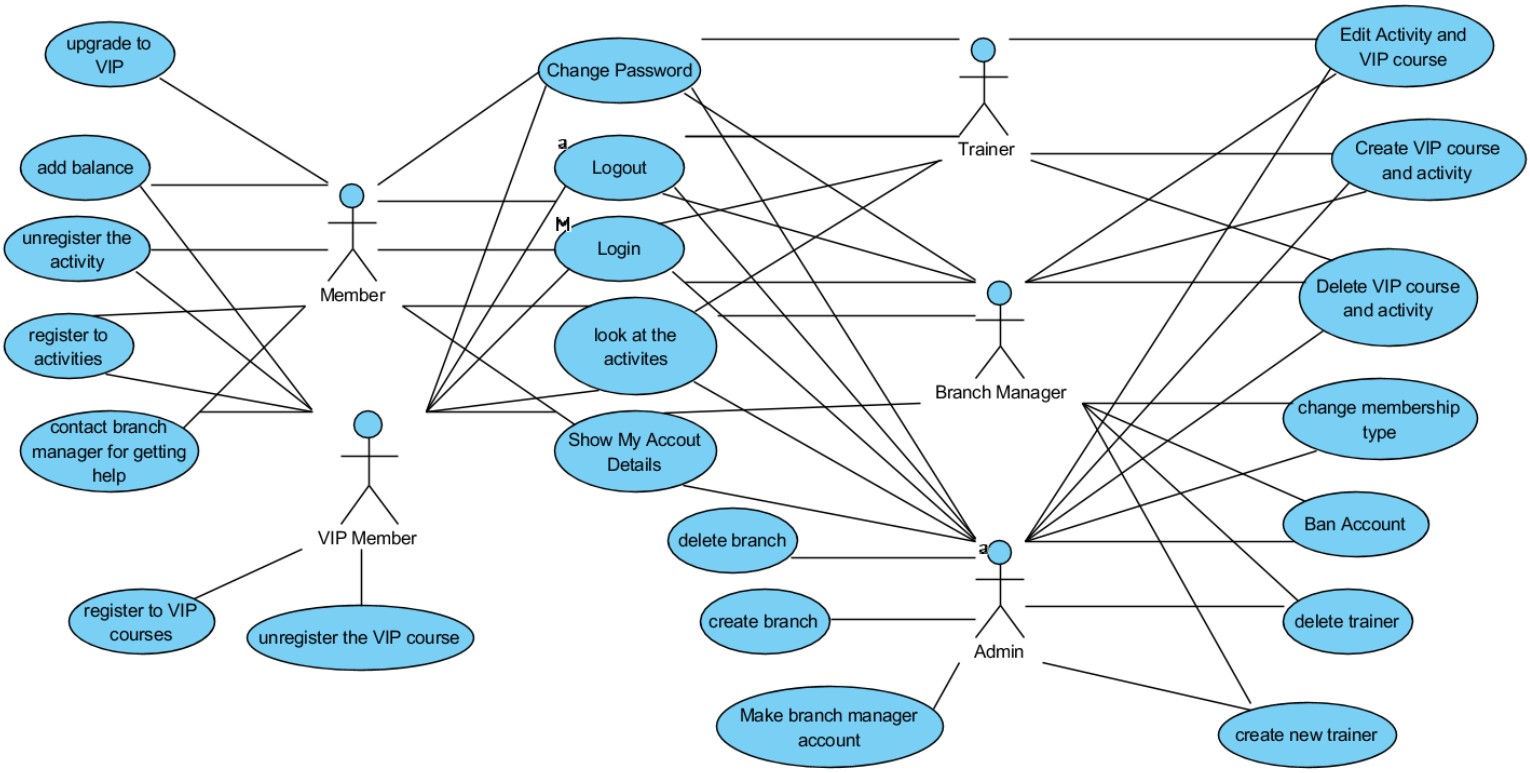
3. Our Configuration / Change Management

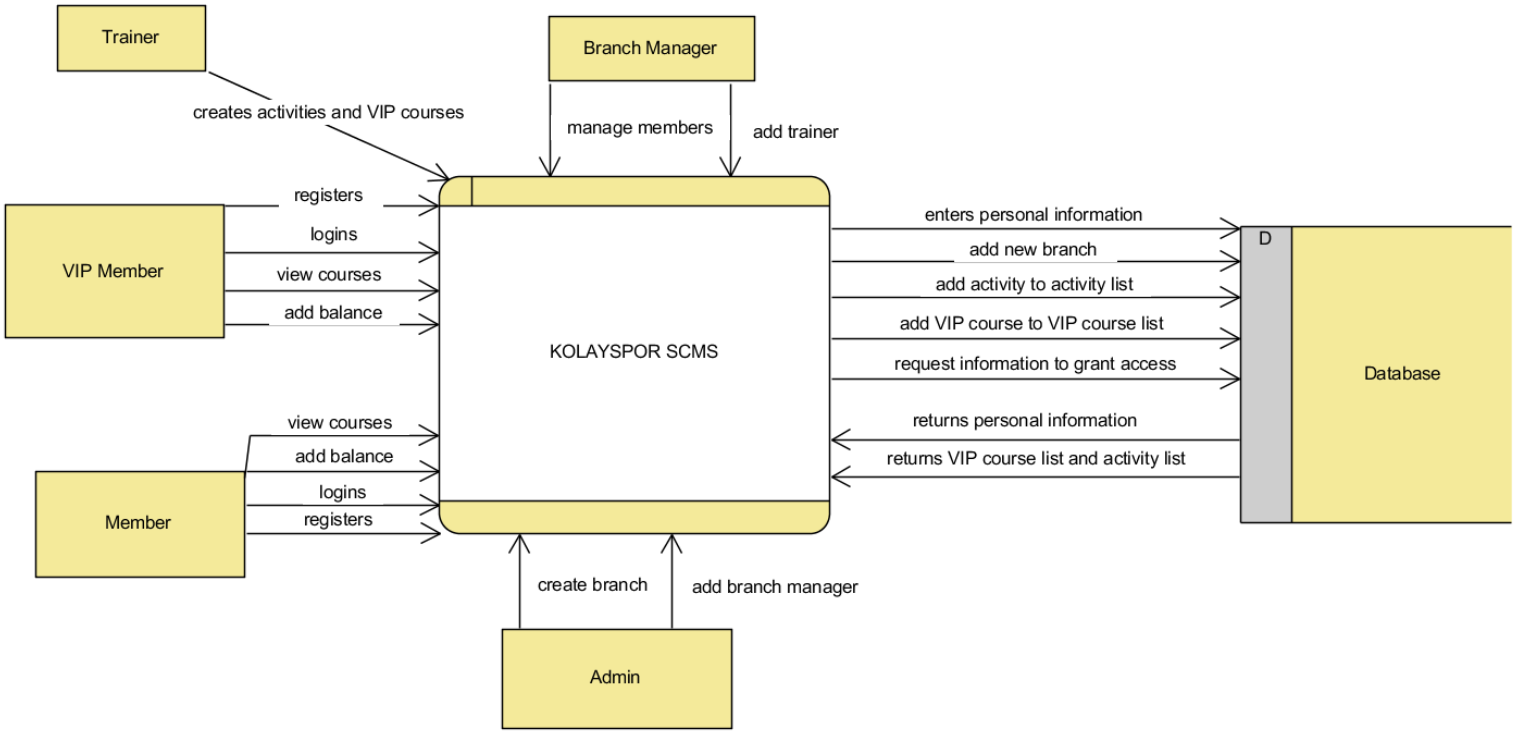
Change management is handled in a social way in our project. When any member of the project has a new idea and wants to change some aspect of the project, he/she will explain his idea and why should it change in the next meeting. All members will discuss the idea considering its pros and cons. At the end of the meeting a final decision can be made or the change request can be suspended until next meeting because the project condition may not be optimal to see future impacts of the change clearly. If the final decision ends up being the change request approved then the required changes are documented and old documents are updated if necessary. Also the schedule is planned about this change.

Configuration management is handled by GitHub. All the older versions of the project is stored there and changes between versions can be viewed. When a developer edits or creates a new module he/she will send it to GitHub with a proper description, explaining what is changed briefly. This helps keeping track of changes. Also developers can work on different parts of the project using branches in GitHub. After the changes are done, the branches can be integrated to the master branch.

Here are the changes in our project:

- We did little change in the use case and context diagrams at the end.





-At first we coded our project with using JSP (JavaServer Pages) but we decided to use spring framework instead of JSP (JavaServer Pages).

-We did not change our easy and clear GUI design.

- We helped each other to hard reports that's why our duties may was increase.

Our policy to make changes;

- The changes do not affect the fulfillment of the requirements
- Try to keep the negative effect at a minimum level in terms of time and cost
- Do not make radical changes on the system
- Appropriate for the purpose of the system
- Does not negatively affect ease of use
- Does not negatively affect performance
- Not cause fault

4. Configuration

We followed a standard cycle of change in the changes we made in this process: