STV Services

Architecture Document

UMKC CS451R Group 8 - Winter 2019

Table of Contents

Change History

Version: 1.0

Modifier: Chale Abad, Mazin Al Ismaili, Ciara Jones, Justin Balino, and Francisca Okojie

Date: 11/08/2019

Description of Change: Initial draft. Contains entire document.

1 Introduction

This STV Services Architecture Document describes the architecture and design for the STV Services application being developed for a group of students' CS capstone class (Group 8) and Professor Bingham. STV Services is a database web application where users can easily search and save networks and/or shows that correspond with their entertainment preferences. From those preferences, specific streaming services like Netflix, Hulu, etc. will be recommended.

The document is designed and released in order to explain the backend and overall structure of the system. The document will go over the design goals, system behaviors and multiple views regarding STV Services. By the end of this document, readers should have a thorough understanding of the creation of the system and how to use it.

The purpose of this document is to describe the architecture and design of the STV Services application in a way that addresses the interests and concerns of all major stakeholders. For this application the major stakeholders are:

- Users and the customer – they want assurances that the architecture will provide for system functionality and exhibit desirable non-functional quality requirements such as usability, reliability, etc.
- Developers – they want an architecture that will minimize complexity and development effort.
- Project Manager – the project manager is responsible for assigning tasks and coordinating development work. He or she wants an architecture that divides the system into components of roughly equal size and complexity that can be developed simultaneously with minimal dependencies. For this to happen, the modules need well-defined interfaces. Also, because most individuals specialize in a particular skill or technology, modules should be designed around specific expertise. For example, all UI logic might be encapsulated in one module. Another might have all business logic.
- Maintenance Programmers – they want assurance that the system will be easy to evolve and maintain on into the future.

The architecture and design for a software system is complex and individual stakeholders often have specialized interests. There is no one diagram or model that can easily express a system's architecture and design. For this reason, software architecture and design is often presented in terms of multiple views or perspectives [IEEE Std. 1471]. Here the architecture of the STV Services application is described from 4 different perspectives [1995 Krutchen]:

1. Logical View – major components, their attributes and operations. This view also includes relationships between components and their interactions. When doing OO design, class diagrams and sequence diagrams are often used to express the logical view.
2. Process View – the threads of control and processes used to execute the operations identified in the logical view.
3. Development View – how system modules map to development organization.

4. Use Case View – the use case view is used to both motivate and validate design activity. At the start of design the requirements define the functional objectives for the design. Use cases are also used to validate suggested designs. It should be possible to walk through a use case scenario and follow the interaction between high-level components. The components should have all the necessary behavior to conceptually execute a use case.

2 Design Goals

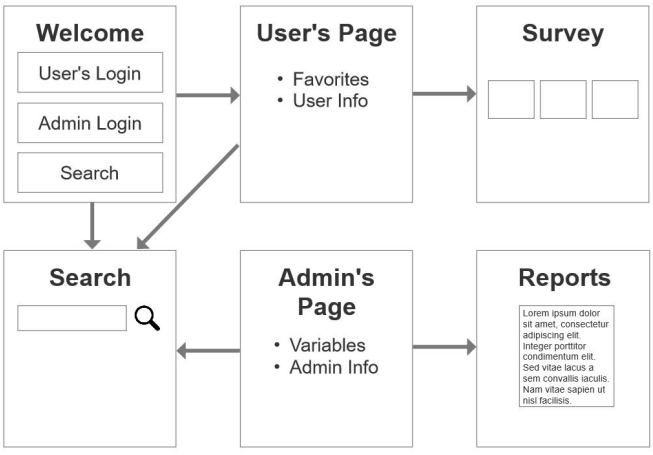
There is no absolute measure for distinguishing between good and bad design. The value of a design depends on stakeholder priorities. For example, depending on the circumstances, an efficient design might be better than a maintainable one, or vise versa. Therefore, before presenting a design it is good practice to state the design priorities. The design that is offered will be judged according to how well it satisfies the stated priorities.

The design priorities for the STV Services application are:

- The design should minimize complexity and development effort.
- The main priorities are usability, efficiency, transparency, and correctness. The users should be able to use the system in a simple and understandable way with correct information.
- The design should take into account that there is a high amount of entertainment. Not all shows, networks, and streaming services will be addressed. The purpose of the assignment is to work on teamwork and a proper logical understanding of the system.
- The design should not focus on maintainability or sustainability. The assignment will be long done and no longer worked on after the semester.

3 System Behavior

The use case view is used to both drive the design phase and validate the output of the design phase. The architecture description presented here starts with a review of the expected system behavior in order to set the stage for the architecture description that follows. For a more detailed account of software requirements, see the requirements document.



4 Logical View

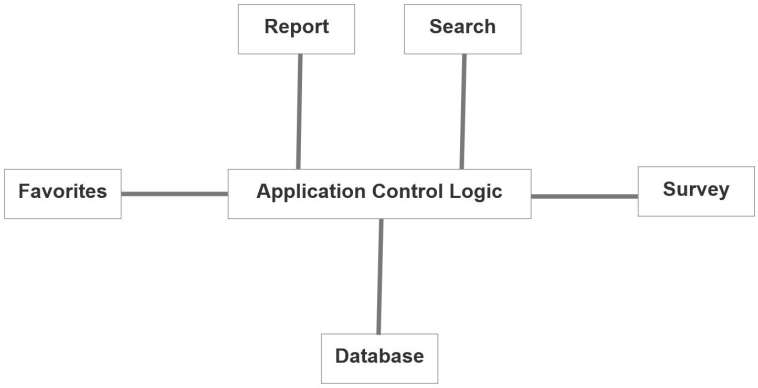
The logical view describes the main functional components of the system. This includes modules, the static relationships between modules, and their dynamic patterns of interaction.

In this section the modules of the system are first expressed in terms of high level components (architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.

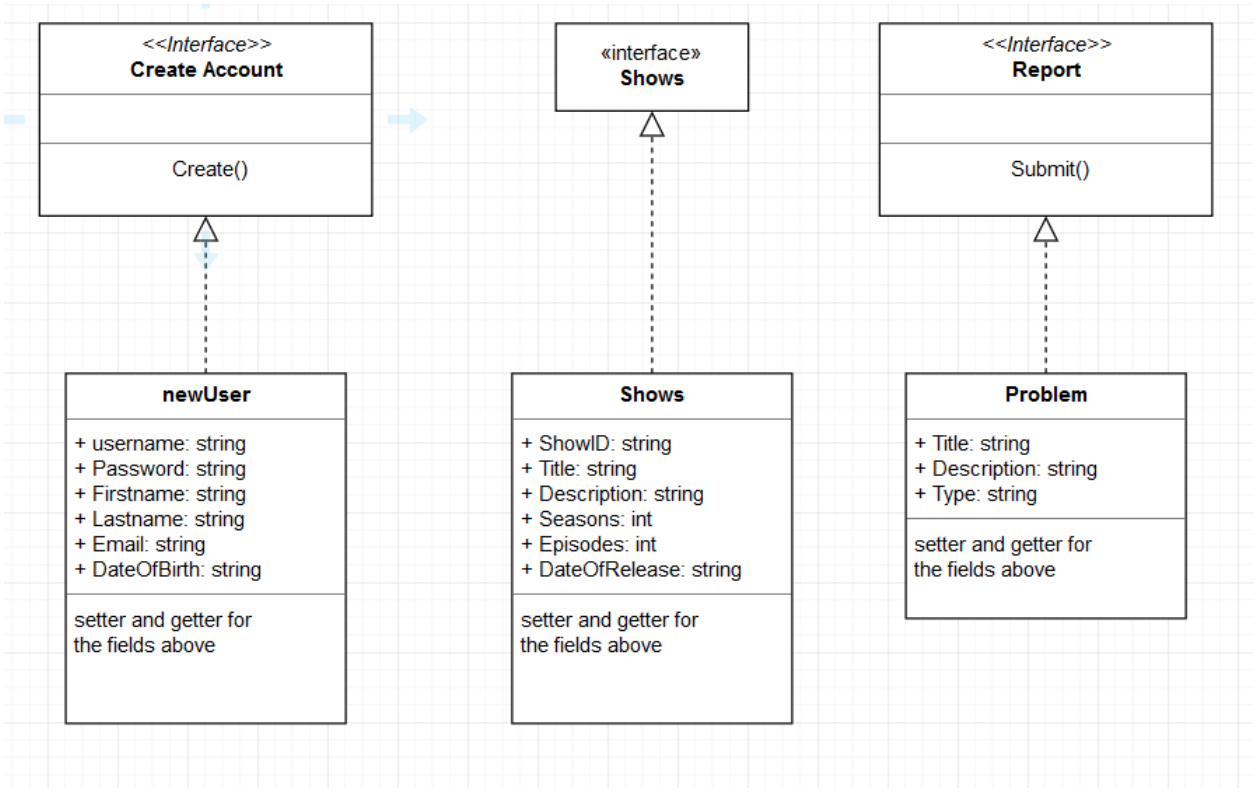
4.1 High-Level Design (Architecture)

The high-level view or architecture consists of 6 major components:

- The **Report** provides the administrator to CREDO user and streaming services information.
- The **Database** is a central repository for data on streaming services, their networks and associated TV shows.
- The **Survey** asks multiple questions regarding entertainment preferences and recommends certain streaming services.
- The **Search** component will allow users to look up certain streaming services, networks, and shows to view their information.
- The **Favorites** list will record users' favorite entertainment preferences corresponding to Search.
- The **Application Control Logic** is the main driver of the application. It presents information to the user and reacts to user inputs.



4.2 Detailed Class Design



5 Physical View

The software elements will be run through Visual Studio C# code and a MySQL database. The assignment does not require actual transfer to hardware. The end product is supposed to simulate a web application that the general public could use.

6 Use Case View