Design and Architecture Guidelines

Architecture and Rationale

After you have completed an initial version of your Vision statement and use cases, you will have a list of customer features expressed in your Product Backlog. You next need to convert this into a design. The first step for your design is to create the Software Architecture for your application. There are many different architectures that you might develop for your system, and you will need to generate a few and then choose between them. You should document the reasons why you choose a particular architecture in your Design Rationale. Rationale is important for the long-term maintenance and enhance of your application. Unlike other programming projects you may have done in other classes, your Senior Design project will live on after you leave the class. This means your design information will be critical to the ability of your customer to maintain and evolve your application over time.

Architecture is the highest, most abstract level of design. It should give a representation of how your application will interact with external entities and how it is organized. We talk about architecture all the time with standard terms like: “Client-Server”, “Layered”, “Event-Driven”, or “Pipe and Filter”. These are commonly called Architectural Styles. Your application may use one or more of these to help you achieve an optimal design for your system.

It is important to provide both a static and dynamic view of your architecture. A static view might, for example, be expressed with a UML Package diagram. A dynamic view could be a system-level sequence diagram or a textual description of how functionality is realized at the architectural level. Each component should be identified as to what functionality it provides and how it interoperates with other components to form a working system. You may use the 4+1 view approach to provide a comprehensive static and dynamic architecture for the system.

Data

Many systems involve the management of persistent data, either through files or a database. How the data is organized is another aspect of design, and its expression is called a data model. If you have taken a database course, then you are already familiar with Entity-Relationship (ER) diagrams. These contain virtually the same concepts as do

CPEN 321 – Software Engineering | Design and Architecture Guidelines 1

UML class diagrams, and the latter can be used if you are unfamiliar with the former. Your design documentation should include one of these visual descriptions of how any persistent data will be stored.

Detailed Design

Decompose the high-level architecture into the lower level classes and document their dynamic behavior. If a non-OO application, document the static and dynamic behavior of the application.

GUI

Many modern systems are interactive. That is, the user interacts with the system using some form of display screen using which data can be entered and results presented. Well- designed GUIs are essential to successful systems because they are the most visible part of the system. As such, it is important to get early and frequent feedback from users about their reactions. This can take the form of a prototype, but if this is infeasible, then mocked up screen shots can be substituted. Your design documentation should include a description of your user interface design.

Validation

Every stage of software development should be validated. This is particularly true of design, because design is the most accessible description of the problem's solution. There are a variety of ways to validate software designs, but all of them involve comparing the design with the user needs. Examples include design reviews with customers, GUI prototypes, and requirements tracing. You should document the steps that you have taken to validate your design and the results of that validation.

**1.Introduction**

Course Planner is to be a website that will enable students to efficiently and easily schedule reminders for all of their coursework. Using information collected from your peers, through a comprehensive course element rating system, as well as custom settings collected from the user, a scheduling system will generate helpful reminders directing you when to begin studying or working on certain assignments. Course Planner is unique to other existing scheduling applications in that it automatically creates and schedules reminders using both user’s preferences and information sourced from other students to allow easier study scheduling.

**Core features:**

1. Edit Courses Schedule

1)Add course

2)Delete course

3)Edit course

1. Reminder System

1) Set a manual reminder

2) Set a rank-based reminder

1. Ranking and Modifying Course Features

1) Rank course feature

2) Add new course feature

3) Add information to existing feature

**Additional features:**

1. Create or login to user profile
2. Use facebook to login to Course Planner profile

2) Create a new course planner profile after first login in with Facebook

**2.Architectural & Component-Level Design**

**Please check the following link to get a clear idea about the important components which need to be included in this part**

**https://www.dropbox.com/s/ule39ewjr0ay3fd/group-3.pdf?dl=1**

**3.Data Design**

**4.User Interface Design**

**a)Login View:**

**b)Schedule Dashboard View:**

**c)Course Information Edit Page View:**

**d)Difficulty Ranking Page View:**

**5.Restrictions, limitations, and constraints**

**6.GUI**

**7.Validation**