*SOFTWARE DESIGN SPECIFICATION*

**1.0 Introduction**

The Software Design Specification (SDS) for the Power Management System. The SDS will break down the project into components to describe in detail what the purpose of each component is and how it will be implemented. The SDS will also serve as a tool for verification and validation of the final product.

**1.1 Goals and objectives**

To achieve software design document, our team can establish exact expectations for the project before starting to code. SDS will help to streamline the code process.

**1.2 Statement of scopes**

The scope of the Power Management System includes its distinct features, its benefits, and its limitations. The system's distinct features allow to view the customer panel, look up electricity usage, pay the pending bills, and look up previous bills. The system enables the user to determine if there are any payments due.  **1.3 Software context**

|  |  |
| --- | --- |
| Term | Definition |
| API | Application Program Interface. An API is a method by which an application program can access the computer's operating system. |
| GUI | Graphical User Interface. The GUI provides a graphical interface for users to interact with the system. |
| Link | A link is a means of connection between screens. Note: This is not short for “Hyperlink.” |
| SDK | Software Development Kit. A set of programs that allows software developers to create products to run on a particular platform or to work with an API. |

**1.4 Major constraints**

Any business or product line constraints that will impact he manner in which the software is to be specified, designed, implemented or tested are noted here.

* The main constraint for this project will be that as we don’t have access to get all the actual bill data from an existing database. So, we will instead work with dummy data inputted by ourselves for this project.

**2.0 Data design**

A description of all data structures and databases.

**2.1 Data structures**

Data structured that are available to major portions of the architecture are described.

**2.2 Database description**

Database(s) created as part of the application is(are) described.

**3.0 Architectural and component-level design**

A description of the software architecture is presented.

**3.1 Architecture diagrams**

Various views (logical, process, physical, development) of architecture are presented with descriptions.

**3.2 Description for Components**

A description of major software components contained within the architecture is presented. Section 3.2.1 is repeated for each of n components.

**3.2.1 Component n description**

**3.2.1.1 Interface description**

Input, output, exceptions, etc.

**3.2.3.2 Static models**

Class diagrams, composite structure diagram, etc.

**3.2.3.3 Dynamic models**

Activity diagrams, sequential diagrams, state diagrams, etc,

**3.3 External Interface Description**

The software's interface(s) to the outside world (other software or hardware systems) are described.

**4.0 User interface design**

A description of the user interface design of the software is presented.

**4.1 Description of the user interface**

A description of user interface including screen images or prototype is presented.

**4.2 Interface design rules**

Conventions and standards used for designing/implementing the user interface are stated.

**5.0 Restrictions, limitations, and constraints**

Special design issues which impact the design or implementation of the software are noted here.

**6.0 Appendices**

Presents information that supplements the design specification.

**6.1 Requirements traceability matrix**

A matrix that traces stated components and data structures to software requirements is developed.

**6.2 Implementation issues**