SOFTWARE DESIGN DESCRIPTION

FOR

SRUTHI - A MALAYALAM VOICE ASSISTANT FOR NATIVE LANGUAGE USERS

Prepared By:

Alen George
Joel Raju
Roshan Roy
San Baby Francis

CONTENTS

1. Overview

1.1 Scope

Sruthi is a voice assistant aiming to help those users whose native language is Malayalam. It will help native language speakers to better understand technology and easily interact with desktop devices. It also helps visually challenged Malayalam speakers to use the features of a computer.

1.2 Purpose

The purpose of this document is to provide a description of the design of the software product to allow for software design to proceed with a perceptive of the design that is to be structured and how the process of it develops. The document give general description of design elements and their interactions, how the system will be structured, data & functional structure are to be further discussed in order to help producing test cases, and help in maintenance services, and also satisfy requirements, design details indicated in the SRS document. This software design document describes the architecture and system design of a voice recognition system that can recognize and respond to speech in Malayalam.

1.3 Intended audience

This website is intended or various purposes as per their requirements. This document intended to be read by project guide, project coordinator and head of computer science department. Reading this document will help the readers sequentially understand the website and its features

1.4 References

- Kurian Benoy, Jiby J Puthiyidam - A Study of Text to Speech Systems for Non-English Languages, IJRAR June 2019, Volume 6, Issue 2
- Preena Johnson, Jishna K C, Soumya S - Speech to Text Conversion in Malayalam, IJLERA July 2017, Volume 2, Issue 7
- Nivedita Singh, Diwakar Yagyasen, Surya Vikram Singh, Gaurav Kumar, Harshit Agrawal – Voice Assistant using Python, IJIRT July 2021, Volume 8, Issue 2

2. Definitions, Acronyms and Abbreviation

BLOCK DIAGRAM	- A diagram showing in schematic form the general arrangement of the parts or components of a complex system or processes.
PC	- Personal Computer
SDD	- Software Design Description
SRS	-Software Requirement Specification
STAKEHOLDERS	- Any person or entity that has an interest in the system.
USER	- A person who uses and interacts with the system.
USER INTERFACE	- An interface that our system contacts with the user of the System, it gets all needed information for its running, from user to our system
STT	- Speech-To-Text

3. Conceptual Model for Software Design Descriptions

This section includes concepts and context of SDD in which the documentation is prepared. The purpose of the conceptual model is to give a better understanding of system terminology and software life cycle that the system resides on. The conceptual model also gives information about stakeholders who will use SDD and how the SDD will be used.

3.1 Software Design in Context

The system will be implemented using python both front end and back end for the backend. The voice assistant will try to give the most accurate result in a most applicable, proper and correct time so it can respond to user's wants correctly and quickly. It should respond correctly and should provide accurate answer of voice command which is requested by the user. The project is designed

in such a way that the user can interact with the system with only using voice as an input.

3.2 Software Design Descriptions within the Life Cycle

This software will be created following the IEEE standards. The primary milestones of the system are requirements analysis, design description analysis, implementation and finally verification and validation.

3.2.1 Influences on SDD Preparation

The very first influence on the software design process is the Sruthi - Malayalam Voice Assistant SRS document. In SDD, we considered the product perspective, functional/nonfunctional requirements and interface requirements that were included in the SRS. Given specifications and the possible new requests from the stakeholders will specify the design process of this system

3.2.2 Influences on Software Life Cycle Products

Before connecting the software and hardware parts of the system, the user interface should be shown with sample examples, which are processed using the interface, to the stakeholders. As a result of this process, stakeholders can share their ideas and requirements about the project. Finally, software and hardware parts will be connected. Furthermore, SDD will guide us all the way through the system. According to this document or the first phase, some requirements can be added or removed from the software requirements. Consequently, requirements of the stakeholders can be met more precisely after each sprint of our development process.

3.2.3 Design Verification and Design Role in Validation

Verification is the process that will test that Sruthi whether it meets a set of design specifications. In this process, we will look at the SRS and SDD documents for correctness of specifications. We will control whether all functional and nonfunctional requirements are correctly implemented according to the requirements of SRS and SDD documents. Furthermore, we will control whether the design viewpoints of the final project are met in the viewpoints part of the SDD document. Validation is the process that the stakeholders and developers decide if the application is consistent with the main goal, which is allowing the

user to use Sruthi as media of communication between system and user. After the complete implementation of the system, the testing process will be handled with SDD influenced test plans and cases.

4. Design Description Information Content

4.1 Introduction

Software description of the Sruthi analyzes how the system will be designed and implemented. This section investigates these according to identification of the SDD, identified design stakeholders and design concerns, related design viewpoints, design views, overlays, rationale and languages. Furthermore, this section includes design elements which are design entities, attributes, relationships and constraints.

4.2 SDD Identification

Sruthi will be released after validation and verification tests. Prototype of the system will be shown in June. Scope, references, context and summary can be found under the section “Overview”. Glossary can be found under the section “Definitions, Acronyms and Abbreviations”

4.3 Design Stakeholders and Their Concerns

Stakeholders comprises the team members of this project along with the faculty. Main concerns of the stakeholders are quality of the system because it must have high quality.

4.4 Design Views

Design views help design stakeholders about focusing on design details from a specific perspective and meeting relevant requirements. Each identified design concern must be the topic of at least one design view so that SDD is complete. Each design concern identified in the previous subsection is the topic of most of the design views in this document; thus, this SDD is completed. For example, concerns about cost are a topic of composition view. Moreover, concerns about the quality of the system are a topic of logical view. In this document, context, composition, logical, dependency, information, patterns use, interface, interaction and state dynamics views will be explained in section 4.5 as their corresponding viewpoints.

4.5 Design Viewpoints

This document describes context, composition, logic, dependency, information, patterns use, interface, structure, and interaction and state dynamics viewpoints.

Context Viewpoint*:* It describes the relationships, dependencies and interactions between the system and its environment such as users and other interacting stakeholders. Interactions between the system and its actors are very intense, hence concerns of this viewpoint are important and suitable for Sruthi, the Malayalam voice assistant. It includes a use case, context and block diagram showing the system boundary.

Composition Viewpoint: It describes how the design subject split up into its components and which roles these components have. It can be used in estimating cost, staffing and scheduling duties of a development team. It includes a deployment and component diagram.

Logical Viewpoint: It describes class structures, interactions between them and how they are designed and implemented. Also, it supports development and reuse of existing logical components. It includes a class diagram which defines objects and classes, and relationships between them.

Dependency Viewpoint: It describes the components of the system and dependencies between these components. It gives information about shared information and order of execution of these components.

Information Viewpoint: It describes data items, data types and classes, data stores and access mechanisms. It gives information about data attributes.

Patterns Use Viewpoint: It describes design patterns and usage of design patterns which meet design ideas of the project.

Interface Viewpoint: It describes the details of external and internal interfaces. It provides information to the designers, programmers and testers before proceeding with the detailed design of the system. This also provides designers, programmers and testers to use the system as a random user.

Interaction Viewpoint: It describes the sequence of actions and how, why, where and at what level actions occur in the system. It is preferred to use state dynamics views in detail for this project.

State Dynamics Viewpoint: It describes the internal behavior of the system. System dynamics include modes, states, transitions and reactions to events. It gives information step by step about the system operation. It includes a state machine diagram which defines conditions, states, transitions and relationships between them.

4.6 Design Elements

Any item which appears in a design view is named as design elements. It may be one or some of these subcases; design entity, design relationship, design attribute and design constraints. All design elements are defined with subcases under their corresponding viewpoint in section 5 of the software design description.

4.7 Design Overlays

Design overlays are usually used to add information to the existing views. This concept will be explained clearly, when necessary, in the design viewpoints in section 5.

4.8 Design Rationale

Object - Oriented approach was chosen while designing because by this way the hardware part and software part will be combined easily. Software part includes parsing and reading notes and transmitting data. To design these lots of packages were used. These packages are connected between each other and they can be controlled separately. Furthermore, for the hardware part a package was used and there is another package to combine software and hardware parts.

4.9 Design Languages

In this document Unified Modeling Language (UML) will be used as the modeling language for the diagrams. The modeling language is used for emphasizing the static structure and dynamic behavior of the system.

5. Design Viewpoints

5.1 Introduction

This section provides several main design viewpoints of Sruthi with their corresponding design concerns and appropriate design languages. Respectively, context, composition, logical, dependency, information, patterns, interface, structure and interaction viewpoints are defined in the following subsections. Short descriptions relating a minimal set of design entities, design relationships, design entity attributes, and design constraints are provided for each viewpoint.

5.2 Context Viewpoint

The context viewpoint of Furniture Sruthi shows the functions provided by a design subject with reference to an explicit context. The services are the functions, which describe the relationships, dependencies, and interactions between the system and its environment like users and other stakeholders.

5.2.1 Design Concerns

Design concerns consist of services, actors and system boundaries. Furniture E-Commerce Website is formed of the interface and the hardware equipment; there are two types of actors which use the system. The services and the system boundaries are established according to the needs of the users. The user can view the products, check aesthetic compatibility with AR and add the products to cart in order to use functionalities like checkout and payment, order customized furniture the user needs to login. Information flow between the user and the website is provided by the interface. Below diagram shows the system boundary which includes the relationship between the user and the other major components of the system.

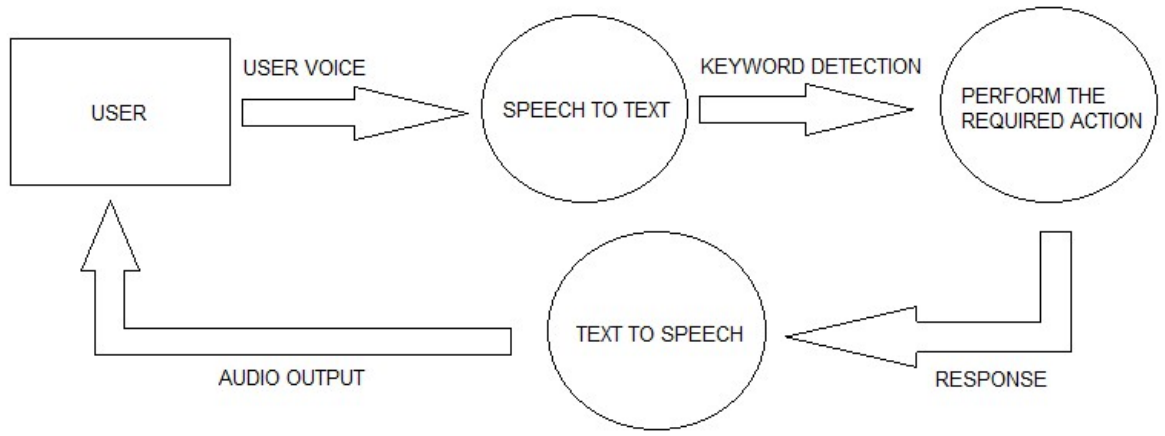

Figure 1: Block diagram of Sruthi

5.2.2 Design Elements

One of the design entities is the actor of the system, user. Stakeholders are other design entities of the voice assistant. The last design entity is the database. Below diagram shows the design relationship which includes provided input and received output between

the user and the other major components of the system. The below diagram describes system interaction and functionalities with users.

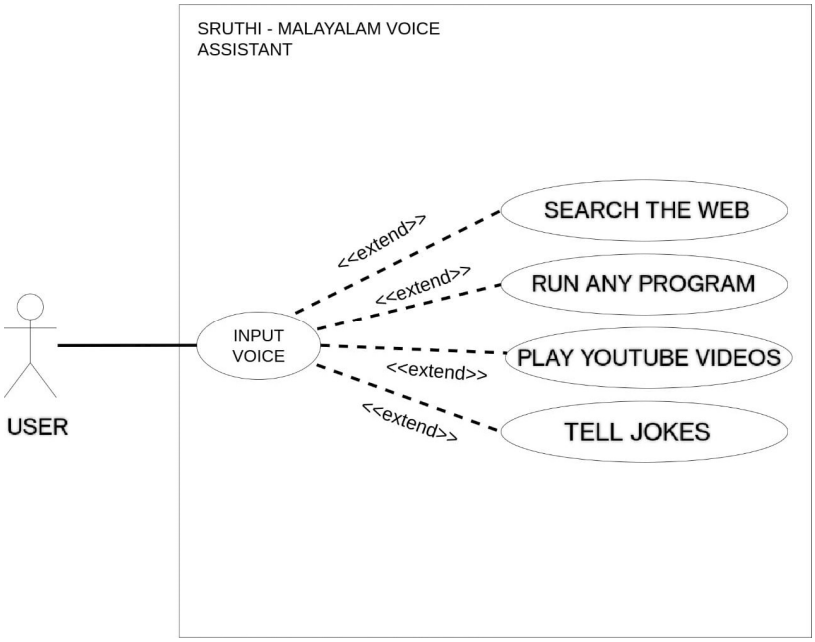

Figure 2: Use case diagram of Sruthi

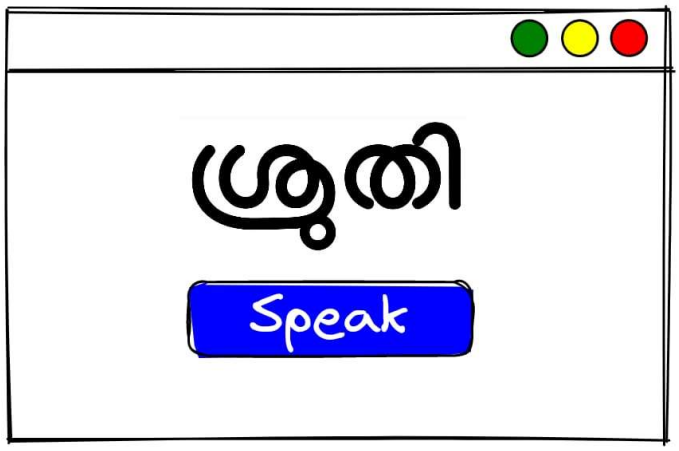

Figure 3: Intended GUI of Sruthi

5.2.3 Example Languages

The diagrams given in the previous subsections are created by the UML. One of these diagrams is the block diagram describing interrelationships of a system. Another one is the context diagram defining the boundary between the system and its environment. The last one of them is the use case diagram showing user interactions with the system.

5.3 Composition Viewpoint

The purpose of the composition viewpoint of Recommendation System is to define the system as a composition of its subsystems. The project is formed by many submodules. Detailed explanation about the relations between these modules will be explained in coming sections.

5.3.1 Design Concerns

With the help of information in composition viewpoint, system stakeholders and developer team can plan and control the software product. The design of this project is structured into sub-modules and components such as interface and other packages. The project is managed by planning, monitoring and controlling these components. All acquired information about the project provides estimated cost, staffing, and schedule for the development effort.

5.3.2 Example Languages

A component diagram showing functional (logical) decomposition of the system and a deployment diagram showing run-time (physical) decomposition of the system have given in the previous sections by using UML modeling language.

5.4 Logical Viewpoint

The purpose of the Logical viewpoint is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships. For each entity, there will be a diagram to overview the entity and then a table that name, return type; visibility of the entity/class diagram is shown in. Also, definitions of each element is provided. After all elements are explained, the class diagram that shows relationships between the classes is drawn.

5.4.1 Design Concerns

The logical viewpoint is employed to show the development and reuse of abstractions and their implementations. This means, object-oriented programming simplifies to maintain and modify existing code while new objects are created. Since identifying object classes is often a difficult part of object-oriented design, during the implementation phase of the project there can be some changes in object identification.

5.4.2 Design Elements

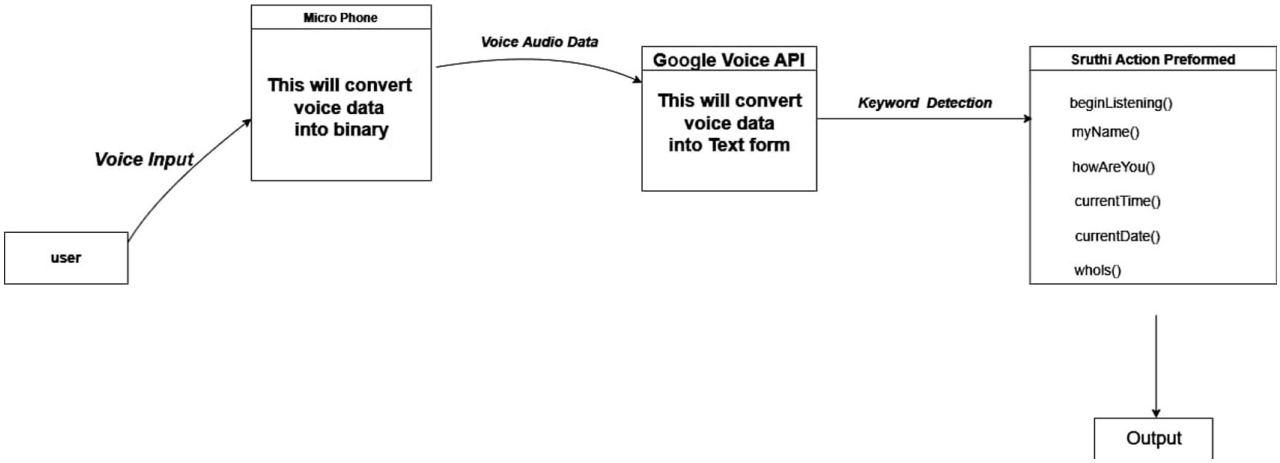

Figure 5: Class Diagram of Sruthi

5.4.3 Example Languages

A class diagram which describes the structure of a system by showing classes of the system has been given in the previous sections by using UML modeling language.