**Software Design Document for "wetest"**

1. Introduction

The purpose of this document is to present the software design for "wetest", a comprehensive mobile application designed to provide users with a holistic learning experience in computer science and related fields. This document outlines the software architecture, modules, interfaces, and data structures used in the application.

2. Architecture Overview

"wetest" is developed as a client-server based application using the Model-View-Controller (MVC) architecture pattern. The server will be responsible for storing user data and handling requests from the client. The client will handle the presentation layer, communicate with the server, and perform business logic operations.

3. Modules

The following modules have been identified for "wetest":

- Authentication module: handles user login and authentication

- Question bank module: manages questions and answers for algorithmic exercises and quizzes

- Knowledge map module: manages knowledge maps that help users understand concepts and their relationships

- Assessment module: provides cognitive assessments and progress tracking

- Coaching module: provides personalized coaching based on user performance

4. Interfaces

The following interfaces will be used in "wetest":

- RESTful API interface: used for communication between the server and client

- User interface: used for presentation and interaction with the user

- Database interface: used to interact with the database for storing and retrieving user data

5. Data Structures

The following data structures will be used in "wetest":

- User profile: stores user information such as name, email, and password

- Questions: stores information about each question, including the text, options, and correct answer

- Knowledge map: stores information about concepts and their relationships

- Cognitive assessment scores: stores user performance data for various assessments

6. Conclusion

The software design presented in this document outlines the architecture, modules, interfaces, and data structures used in "wetest". By utilizing a client-server architecture pattern and the MVC design pattern, we aim to provide a robust and scalable application that delivers a personalized learning experience for users.