

System Requirement **Specifications**

“Programming Assistant Web App”

Guidance by -
Prof. Amit Kumar sir

Project by -
Ganesh Singh - U101115FCS087 - S5
Sangamesh Kotalwar - U101115FCS210 - S5
Shivam Ratnakar - U101115FCS147 - S5
Shivangi Tak - U101115FCS213 - S5

Software Requirement Specification

“Programming Assistant Web App”

1. Introduction

1.1. Purpose:

We all go through the trouble of googling about any function, data type or syntax while coding while learning a new programming language.

Programming Assistant is a Web App where the user can search a keyword/task and get all the help about it with a code snippet and description, in whichever language user chooses from the ones provided by the system.

The user can even ask questions like how can we validate a password and get the description and code snippet for the same. It can be used by him/her while coding! The search time is also reduced as when you search on Google, it provides references to multiple sites to get a single working solution!

This app also provides the user (who must be logged in) with the option to save all their codes and functions made by them which will work as a guide for them in future.

The user can even share his snippets with his friends as well as publically (i.e. making it visible to all registered users).

1.2. Scope:

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In scope:

1. User will be able to find all the assistance he needs regarding the entered coding language on any entity or operation and also save (if registered) his experiences as code snippets for future reuse.
2. Creating and managing a personal account of his own.
3. The registered user will be able to store all his learning and building experiences at one place along with their description and retrieve them from anywhere in future.
4. (if registered) User will be able to share his learning and building experiences with any other existing user he wants.
5. The code snippets stored are in sorted order according to the tags and category so that the searching time is less.
6. User will be able to specify whether the snippet he is entering is to be kept private or public.

Out of scope:

1. Any user cannot know about the private code snippets of the other users and request for it.
2. The entered code snippet or description cannot be checked for correctness.
3. Users can't rate one another's public data for better references.

4. User won't be able to get the detailed information/documentation about the entity searched by him/her.

1.3. References

www.geeksforgeeks.org
www.w3schools.com
<https://developer.mozilla.org/>
<https://stackoverflow.com/>
<https://github.com/>

1.4 Overview

The rest of this SRS is organized as follows:

Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed.

Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.

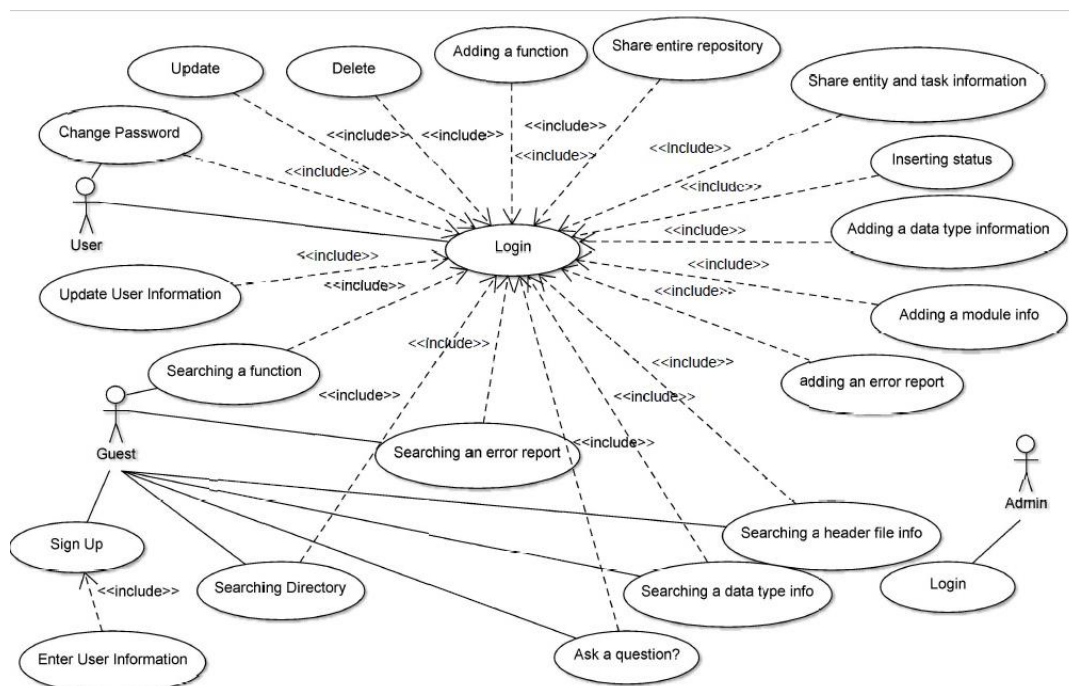
Section 4 gives some possible future extensions of the system.

Finally the appendices in Section 5 describe where we have found the data from for our database.

2. Overall description

2.1. Product Perspective

The main motto of Programming Assistant system is to allow any programmer irrespective of his knowledge and experience, to store (only registered users) and search all his learning and building experiences and being able to share them with other people who can use that knowledge to increase theirs. The main motive is that the user does not have to search for the same thing twice.



2.2 Product Functions

PA should support the following use cases:

Class of use cases	Use cases	Description of use cases
Use cases related to Application Access	Access as a guest	Guest access to the system's database
	Access as a user	Access to system's database as well as other users public snippets and private files of the users
Use cases related to authorization	Login	Login into PA
	Change Password	Change PA password
	Sign up	Create your account by entering unique email id and password.
Use cases related to creating profile	Enter user information	Takes user information
	Update user information	Updates user information
Use cases related to searching	Searching a function	Search for functions.
	Searching an error report	Search for error report.
	Searching a header file info.	Search for header file.
	Searching a data type info.	Search for data type.
	Ask a question	Ask a question about how to do a certain task.
	Searching Directory?	Search in personal directory or general public search.
Use cases related to sharing	Share entity or task information	Sharing entity or task information with another user
	Share entire Directory	Sharing entire Directory with another request
Use cases related to directory	Inserting a function	Add a new function to current personal directory.
	Inserting an error report	Add a new error to current personal directory.
	Inserting a header file info.	Add a new header file to current personal directory.
	Inserting a data type info.	Add a new data type to current personal directory.
	Inserting status	Insert in directory as public access or personal access.
	Updating information	Updates an entity
	Deleting information	Deletes an entity

2.3. User Characteristics

- a. User should be able to frame the description and keywords for the searches and inserts he wishes to perform.
- b. The user should have a good internet access.
- c. User should be familiar with the coding languages he is using

2.4. Principal Actors

Principal actor is "user" as well as our "admin".

2.5. General constraints

For a proper and full working, internet access is required.

2.6. Assumptions and Dependencies

1. Full working is dependent on the availability of internet connection.
2. The code snippets entered by the user in his directories are working.

3. Specific requirements

3.1. Functional requirements

Use cases related to Application Access

Use Case 1: Access as a guest

Primary Actor: User

Pre-Condition: User should be on the homepage of the website.
Internet connection should be established.

Main Scenario:

1. User chooses to directly search without logging in or signing up.
2. Results are shown on the basis of system hits in the App's Database(default database)

Alternate scenario:

3. Network failure
- 3.(a) Search aborted.

Use Case 2: Access as a user

Primary Actor: User

Pre-Condition: User should be on the homepage of the website.
Internet connection should be established.

Main Scenario:

1. User chooses to log in or sign up.
2. User's profile opens up.
3. For search: Access to system's database as well as other public snippets of other users.

4. Can share his public files with other users.
5. Insert snippets, update or delete existing snippets.

Alternate scenario:

6. Network failure
 - 6.(a) Action Aborted.

Use cases related to authorization:

Use Case 3: Login

Primary Actor: User

Pre-Condition: There must be internet connection and login webpage must be opened in the browser.

Main Scenario:

1. Open the webpage in browser. User prompted for login and password.
2. User gives the login and password.
3. System does authentication.
4. Main screen is displayed.

Alternate Scenario:

- 5(a). Authorization fails
 - 5(a) 1 Prompt the user that he typed the wrong password.
 - 5(a) 2 Allow him to re-enter the password. Give him 3 chances.

Use Case 4: Change Password

Primary Actor: User

Pre-Condition: User logged in and there must be internet connection

Main Scenario:

1. User initiates the password change command.
2. User is prompted for old password, new password and confirm new password.
3. User gives the old password, new password and confirm new password.
4. System does authentication.
5. New password is registered with the system.

Alternate Scenario:

- 6(a). Authorization fails
 - 6(a) 1 Prompt the user that he typed the wrong password.
 - 6(a) 2 Allow him to re-enter the password. Give him 3 chances.
- 6(b) New password and confirm new password do not match.
 - 6(b) 1. Allow him to re-enter the attributes. Give 3 chances.

Use Case 5: Sign up

Primary Actor: User

Pre-Condition: There must be internet connection and home webpage must be opened in the browser.

Main Scenario:

1. Open the home webpage in browser. User prompted for sign up.
2. User prompted for entering email-id and at the same time notify the user if an account with the same email id exists and prompt him to click verify button.
3. System sends verification code to entered email id and prompts user to enter the verification code. System also provides the user with the option to resend the code in case it was not received.
4. System verifies the code.
5. Main screen is displayed.

Alternate Scenario:

6(a). Authorization fails

6(a) 1. Prompt the user that he typed the wrong code.

6(a) 2. Allow him to re-enter the code. Give him 3 chances.

Use cases related to creating profile

Use Case 6: Enter user information

Primary Actor: User

Pre-Condition: There must be internet connection and the user must be logged into the app.

Main Scenario:

1. Log into the web app. User prompted for updating the profile information if any of the field is unfilled
2. Prompt user to enter his full name and age and mark it as necessary field.
3. Prompt user to enter his job title and description and skills possessed.
4. Prompt user to enter which languages he codes in and mark it as a necessary field.
5. Provide user with a Done button if all the fields are filled.

Alternate scenario:

6. Network failure

6.(a) Action Aborted.

Use Case 7: Update user information

Primary Actor: User

Pre-Condition: There must be internet connection and the user must be logged into the app.

Main Scenario:

1. Log into the web app. Main screen is displayed with an update profile button in the corner of the page if all the fields in the profile is filled.

2. Display the user information page if update button is clicked.
3. Allow him to re-enter or edit any of the fields he wishes.
4. Prompt user to click on the save button after making all the changes.

Alternate scenario:

6. Network failure
- 6.(a) Action Aborted.

Use cases related to searching

Use Case 8: Searching for a function

Primary Actor: User

Precondition: At least some knowledge of how to enter search strings.

Main Scenario:

1. User first needs to select the filter options provided (buttons) so that he gets exactly what he wants.
2. Enter the search string.
3. Press the search button.

Alternate Scenario:

- 4(a) The user will get lots of unnecessary results if filters aren't used properly.

Use Case 9: Searching for an error report

Primary Actor: User

Precondition: At least some knowledge of how to enter search strings.

Main Scenario:

1. User first needs to select the filter options provided (buttons) so that he gets exactly what he wants.
2. Enter the search string.
3. Press the search button.

Alternate Scenario:

- 4(a) Display of unnecessary results if filters aren't used properly.

Use Case 10: Searching for a header file

Primary Actor: User

Precondition: At least some knowledge of how to enter search strings.

Main Scenario:

1. User first needs to select the filter options provided (buttons) so that he gets exactly what he wants.
2. Enter the search string.
3. Press the search button.

Alternate Scenario:

4(a) The user will get lots of unnecessary results if filters aren't used properly.

Use Case 11: Searching for a data type

Primary Actor: User

Precondition: At least some knowledge of how to enter search strings.

Main Scenario:

1. User first needs to select the filter options provided (buttons) so that he gets exactly what he wants.
2. Enter the search string.
3. Press the search button.

Alternate Scenario:

4(a) Display of unnecessary results if filters aren't used properly.

Use Case 12: Ask a question (For eg: How to..?)

Primary Actor: User

Precondition: At least some knowledge of how to enter search strings.

Main Scenario:

1. User first needs to select the filter options provided (buttons) so that he gets exactly what he wants.
2. Enter the question which proper keywords.
3. Press the search button.

Alternate Scenario:

4(a) Display of unnecessary results if filters aren't used properly.

Use Search 13: Search Directory

Primary Actor: User

Precondition: None

Main Scenario:

1. User first needs to select the filter options provided (buttons) so that he gets from exactly where he wants i.e. he wants to search his own directory or he wants a general search in all user public directories.
2. Enter the search string.
3. Press the search button.

Alternate Scenario:

4(a) Display of unnecessary results if filters aren't used properly.

Use cases related to sharing

Use Case 14: Share entity or task information

Primary Actor: User

Pre-Condition: There must be internet connection and the user must be logged into the app.

Main Scenario:

1. Log into the web app. User prompted to click the SHARE button for sharing a code snippet or entire Directory with another user.
2. Prompt user to select the type of sharing to be done (sharing a particular code snippet or sharing the entire Directory).
- 3 Prompt user to select the category under which the search is to be performed (entity or task) if sharing a code snippet is selected as the category.
4. Allow user to search for the code snippet in his category.
- 5 If code snippet is found prompt user to enter the user_id with which he wants to share the code snippet and at same time check whether the user_id exists or not and notify the user with an error if user_id does not exist and allow him to re-enter the user_id.
6. Prompt user to click the SHARE button if all the fields are shared correctly.

Alternate scenario:

7. (a) If code snippet not found.
 - 7(a).1 Notify user with a message that code snippet not found and allow him to run the search again.

Use case 15: Share entire Directory

Primary Actor: User

Pre-Condition: There must be internet connection and the user must be logged into the app

Main Scenario:

1. Log into the web app. User prompted to click the SHARE button for sharing a code snippet or entire Directory with another user.
2. Prompt user to select the type of sharing to be done (sharing a particular code snippet or sharing the entire Directory).
- 3 Prompt user to select the category under which the search is to be performed (entity or task) if sharing entire Directory is selected as the category.
4. Prompt user to enter the user_id with which he wants to share the Directory and at same time check whether the user_id exists or not and notify the user with an error if user_id does not exist and allow him to re-enter the user_id.
5. Prompt user to click the SHARE button if all the fields are shared correctly.

Use cases related to adding information into own Directory

Use case 16: Inserting a function

Primary Actor: User

Precondition: User must be logged in. Activate internet connection.

Main Scenario:

- 1 Once the user has something new to store for future reference, he will select proper filters (buttons or options) of what kind of information he has which he wants to add.
2. Click the 'ADD' button.

Alternate Scenario:

3(a) Information not entered properly

- 3(a)1. The user or any other user will have difficulties accessing the info again if not added properly.

Use Case 17: Adding error report

Primary Actor: User

Precondition: User must be logged in. Activate internet connection.

Main Scenario:

1. Once the user has something new to store for future reference, he will select proper filters (buttons or options) of what kind of information he has which he wants to add.
2. Click the 'ADD' button.

Alternate Scenario:

3(a) Information not entered properly

- 3(a)1. The user or any other user will have difficulties accessing the info again if not added properly.

Use Case 18: Inserting header file info

Primary Actor: User

Precondition: User must be logged in. Activate internet connection.

Main Scenario:

1. Once the user has something new to store for future reference, he will select proper filters (buttons or options) of what kind of information he has which he wants to add.
2. Click the 'ADD' button.

Alternate Scenario:

3(a) Information not entered properly

- 3(a)1. The user or any other user will have difficulties accessing the info again if not added properly.

Use Case 19: Inserting Data type info

Primary Actor: User

Precondition: User must be logged in. Activate internet connection.

Main Scenario:

1. Once the user has something new to store for future reference, he will select proper filters (buttons or options) of what kind of information he has which he wants to add.
2. Click the 'ADD' button.

Alternate Scenario:

- 3(a) Information not entered properly
 - 3(a)1. The user or any other user will have difficulties accessing the info again if not added properly.

Use Case 20: Inserting Status

Primary Actor: User

Precondition: None

Main Scenario:

1. Once the user clicks the add button, he will be asked a last question whether he wants store his information as Public or Private.
2. Click the 'PUBLIC' or 'PRIVATE' button.

Alternate Scenario:

- 3(a) Information not entered properly
 - 3(a)1. The user or any other user will have difficulties accessing the info again if not added properly.

Use case 21: Updating information

Primary Actor: User

Pre-Condition: There must be internet connection and the user must be logged into the app.

Main Scenario:

1. User can insert new information or update existing information.

Alternate scenario:

- 2(a). If the code snippet is not found
 - 2(a).1 Allow user to run the search again
- 2(b) Network error
 - 2(b)1. Action aborted.

Use case 24: Deleting information

Primary Actor: User

Pre-Condition: There must be internet connection and the user must be logged into the app.

Main Scenario:

1. User opens his Directory.
2. Searches for the required code, clicks on delete button. (if found)

Alternate scenario:

3 (a). If the code snippet is not found

3 (a).1 Allow user to run the search again

3.2. Non-Functional Requirements

3.2.1 Performance Requirements

1. Should have a proper internet connection.
2. Existing account on the web app.
3. Any functional web Browser.

3.2.2 Security Requirements

1. Session multiplexing can be an issue because many users can login, store, search and share content at the same time therefore managing sessions and cookies is very important.
2. Make sure other spam websites don't affect our web app. Also non validated redirects and forwards should not happen.

3.2.3 Business Rules

1. None of the admin has the right to share the code publicly.
2. Admin can delete any content or account if required.

3.4 External Interface Requirements

None.

4. Future Extensions

It is multi-user web app, future extensions may include functionalities like sending share requests to other users, if a user wants something from another user. A voting functionality can also be introduced so that the search results will give only highly rated snippets as output.