****

**COMSATS University Islamabad**

**Abbottabad, Pakistan**

**Interview Preparation Application**

***By***

**Hozefa Hassan Rizvi CIIT/FA20-BSE-019/ATD**

**Muhammad Hammad CIIT/ FA20-BSE-031/ATD**

**Ashar Ali CIIT/ FA20-BSE-158/ATD**

***Supervisor*Mam Sana Malik**

***Bachelor of Science in Computer Science (2020-2024)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

****

**COMSATS University, Islamabad Pakistan**

**Interview Preparation Application**

**A project presented to**

**COMSATS Institute of Information Technology, Islamabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Computer Science (2020-2024)***

**By**

**Hozefa Hassan Rizvi CIIT/FA20-BSE-019/ATD**

**Muhammad Hammad CIIT/ FA20-BSE-031/ATD**

**Ashar Ali CIIT/ FA20-BSE-158/ATD**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Hozefa Hassan Rizvi Muhammad Hammad Ashar Ali

--------------------------- --------------------------- ---------------------------

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (SE) “Interview Preparation Application” was developed by **Hozefa Hassan Rizvi (CIIT/FA20-BSE/SE-019)** and **Ashar Ali(CIIT/FA20-BSE/SE-158)** and **Muhammad Hammad (CIIT/FA20-BSE/SE-031)** under the supervision of “Mam Sana Malik” her opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Software Engineering.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**EXECUTIVE SUMMARY**

The **"Interview Preparation Application"** is a special application designed to help software engineering students who recently graduated prepare for job interviews. It is made to use the knowledge and experiences shared by users. The app allows sharing questions and answers. Users can read expert opinions and success stories of others. The mock interview section analyzes gestures and facial e­xpressions without needing the­ right answers. This unique way builds confidence. Content quality is important, so users validate it together as a community. As a nonjudging space, the main goal is to increase user confidence. This makes the app very helpful for new software engineering graduates entering the competitive job market.

**ACKNOWLEDGEMENT**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mam Sana Malik”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Hozefa Hassan Rizvi Muhammad Hammad Ashar Ali

--------------------------- --------------------------- ---------------------------

**ABBREVIATIONS**

|  |  |
| --- | --- |
| **SRS** | Software Require Specification |
| **SD** | Sequence Diagram |
| **UML** | Unified Modeling Language |
| **App** | Application |
| **AI** | Artificial Intelligence |
| **OpenCV** | Open Cloud Vision |
| **SDD** | Software Design Document |

**TABLE OF CONTENTS**

[**1 Introduction** 11](#_Toc520773734)

[1.1 Brief 11](#_Toc520773735)

[1.2 Relevance to Course Modules 11](#_Toc520773736)

[1.3 Project Background 11](#_Toc520773737)

[1.4 Literature Review 11](#_Toc520773738)

[1.5 Analysis from Literature Review (in the context of your project) 11](#_Toc520773739)

[1.6 Methodology and Software Lifecycle for This Project 11](#_Toc520773740)

[1.6.1 Rationale behind Selected Methodology 11](#_Toc520773741)

[**2** **Problem Definition** 12](#_Toc520773742)

[2.1 Problem Statement 12](#_Toc520773745)

[2.2 Deliverables and Development Requirements 12](#_Toc520773746)

[2.3 Current System (if applicable to your project) 12](#_Toc520773747)

[**3** **Requirement Analysis** 13](#_Toc520773748)

[3.1 Use Cases Diagram(s) 13](#_Toc520773750)

[3.2 Detailed Use Case 13](#_Toc520773751)

[3.3 Functional Requirements 13](#_Toc520773752)

[3.4 Non-Functional Requirements 13](#_Toc520773753)

[**4 Design and Architecture** 14](#_Toc520773754)

[4.1 System Architecture 14](#_Toc520773759)

[4.2 Data Representation [Diagram + Description] 14](#_Toc520773760)

[4.3 Process Flow/Representation 14](#_Toc520773761)

[4.4 Design Models [along with descriptions] 14](#_Toc520773762)

[**5** **Implementation** 15](#_Toc520773763)

[5.1 Algorithm 15](#_Toc520773765)

[5.2 External APIs 15](#_Toc520773766)

[5.3 User Interface 15](#_Toc520773767)

[**6 Testing and Evaluation** 16](#_Toc520773768)

[6.1 Manual Testing 16](#_Toc520773771)

[6.1.1 System testing 16](#_Toc520773772)

[6.1.2 Unit Testing 16](#_Toc520773773)

[6.1.3 Functional Testing 17](#_Toc520773774)

[6.1.4 Integration Testing 17](#_Toc520773775)

[6.2 Automated Testing: 18](#_Toc520773776)

[Tools used: 18](#_Toc520773777)

[**7** **Conclusion and Future Work** 19](#_Toc520773778)

[7.1 Conclusion 19](#_Toc520773780)

[7.2 Future Work 19](#_Toc520773781)

[**8** **References** 20](#_Toc520773782)

**LIST OF FIGURES**

Fig 1.1 Block Diagram 8

Fig 2.1 Use Case Diagram 9

**LIST OF TABLES**

Table 1.1 Table1 8

Table 2.1 Table 2 9

1. **Introduction**

This chapter provides a comprehensive overview of the "Interview Preparation Application" project. To begin, this paragraph summarizes the key aspects covered in this chapter.

* 1. **Brief**

In this brief yet crucial section, the project's essence is encapsulated. The Interview Preparation Application stands as a valuable tool for graduated software engineering students, offering a unique blend of collaborative learning and confidence-building techniques for interview preparation. The introduction touches upon the tools and methodologies employed in crafting this application, providing a glimpse into the upcoming chapters' discussions on its development, distinctive features, and the innovative approach taken towards mock interviews. This chapter sets the foundation for a detailed exploration of the project's intricacies and outcomes.

* 1. **Relevance to Course Modules**

We have studied React Native, JavaScript, Requirement, Software Development at the university before, but we were not familiar with those concepts. Then afterward we studied the course Mobile Application development, Website Technologies, Software Requirements Engineering, Software Quality Engineering, Software Design and Architecture, and Design Patterns. All these courses make us able to understand the process behind any software system. And all these courses made us able to think and design and develop better applications. And then the course of Mobile Application Development guides us to choose the language to develop the software which we were willing to do

* 1. **Project Background**

The Interview Preparation Application emerges from the realization that traditional interview prep methods lack a tailored approach for graduated software engineering students. This project seeks to address this gap by creating a collaborative platform, encouraging knowledge sharing and real-time interactions. The unique feature of analyzing user gestures in mock interviews adds an innovative dimension, aiming to enhance overall interview readiness. The project's core idea revolves around empowering users through a dynamic, community-driven approach to interview preparation.

* 1. **Literature Review**

The current literature reveals a shift towards personalized and interactive interview preparation tools, emphasizing community-driven platforms for shared learning among graduated software engineering students. Dynamic content, such as question banks and expert opinions, is gaining prominence. Additionally, there is a growing interest in utilizing technology to analyze non-verbal cues during interviews. The Interview Preparation Application aligns with these trends, providing a holistic solution that combines collaborative learning with innovative features, contributing to the evolving landscape of interview preparation tools for software engineering graduates.

* 1. **Analysis from Literature Review**

We tried to use OpenCV, deep face for mock interview module to the user by using its front camera, just to make it interesting and interactive. We have used this technology to make the interview look realistic and not look like a video on YouTube. We have provided the confidence level tracking feature to make it more interactive. It will look like a user is giving an interview in real life.

* 1. **Methodology and Software Lifecycle for This Project**

The methodology employed for this project revolves around a procedural design approach, tailored to the specific demands of the React Native technology stack. React Native predominantly follows a procedural design method, emphasizing step-by-step execution and the use of functions for streamlined development. This choice ensures compatibility and optimal performance within the framework, aligning with the nature of JavaScript.

In terms of the Software Development Life Cycle (SDLC) model, the project has embraced the Agile methodology. Agile's iterative development approach, flexibility, and continuous user feedback capabilities make it particularly suitable for this project. The dynamic nature of the project requirements necessitates adaptability throughout the development process, and Agile's collaborative, user-centric focus has been instrumental in guiding the project through its various stages. The adoption of Agile ensures efficient and responsive development, ultimately contributing to the project's ongoing success.

* + 1. **Rationale behind Selected Methodology**

The approach of designing primarily in line with React Native and avoiding strict adherence to Object-Oriented (OO) principles comes from the nature of our project requirements While we primarily work with JavaScript, React Native works well has a planning process that emphasizes step-by-step implementation and a workflow. This approach ensures efficiency and consistency across our chosen technology stack.

Regarding the software development life cycle (SDLC), the choice of Agile methodology depends on its suitability for the dynamic and evolving nature of our project. Unlike a rigid OO approach, Agile allows for iterative improvement, ensures flexibility, and incorporates continuous user feedback. This flexibility is important for a project like ours, where changing requirements are common, and the Agile methodology’s concept of collaborative processing fits well with our objectives Overall, the approach we have chosen is an SDLC model is a practical choice to suit the unique needs and nature of our business.

A diagram of a design process

Description automatically generated

*Figure 1.1: Iterative SDLC*

1. **Problem Definition**

4. 1. **Problem Statement**

The "Interview Preparation Application" handles the challenges faced by software engineering graduates when getting ready for job interviews. Graduates often miss out on job opportunities and feel stressed because they haven't learned how to navigate the real-world job scene. Universities and colleges usually don't teach students what to expect in actual job interviews and how to behave in a professional setting.

* 1. **Deliverables and Development Requirements**

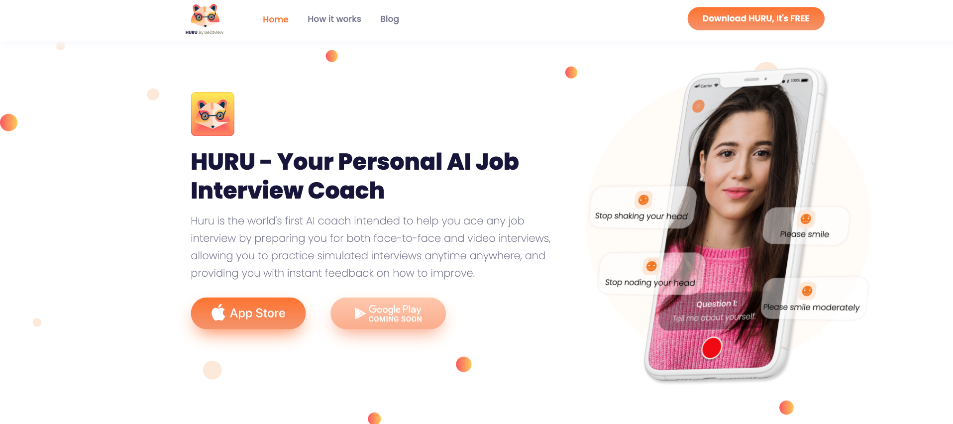
The anticipated deliverables for this project include:

* A fully functional Interview Preparation Application catering to graduated software engineering students.
* User authentication and profile setup features.
* Integration of a comprehensive question bank for interview preparation.
* Real-time collaboration capabilities with experts.
* A mock interview section that evaluates user confidence through expressions.

Development requirements encompass:

* Implementation of a secure authentication system.
* User-friendly interface design for seamless navigation.
* Integration of a firestore database to store user profiles, questions, and responses.
* Incorporation of group chat feature for expert collaboration.
  1. **Current System**

The existing system, Huru AI, provides mock interview services but operates on a paid subscription model. Users are required to pay a fee to access the mock interview feature, creating a financial obstacle for some graduated software engineering students. This limitation poses a challenge for individuals seeking frequent and affordable interview practice, hindering their ability to build confidence and experience.

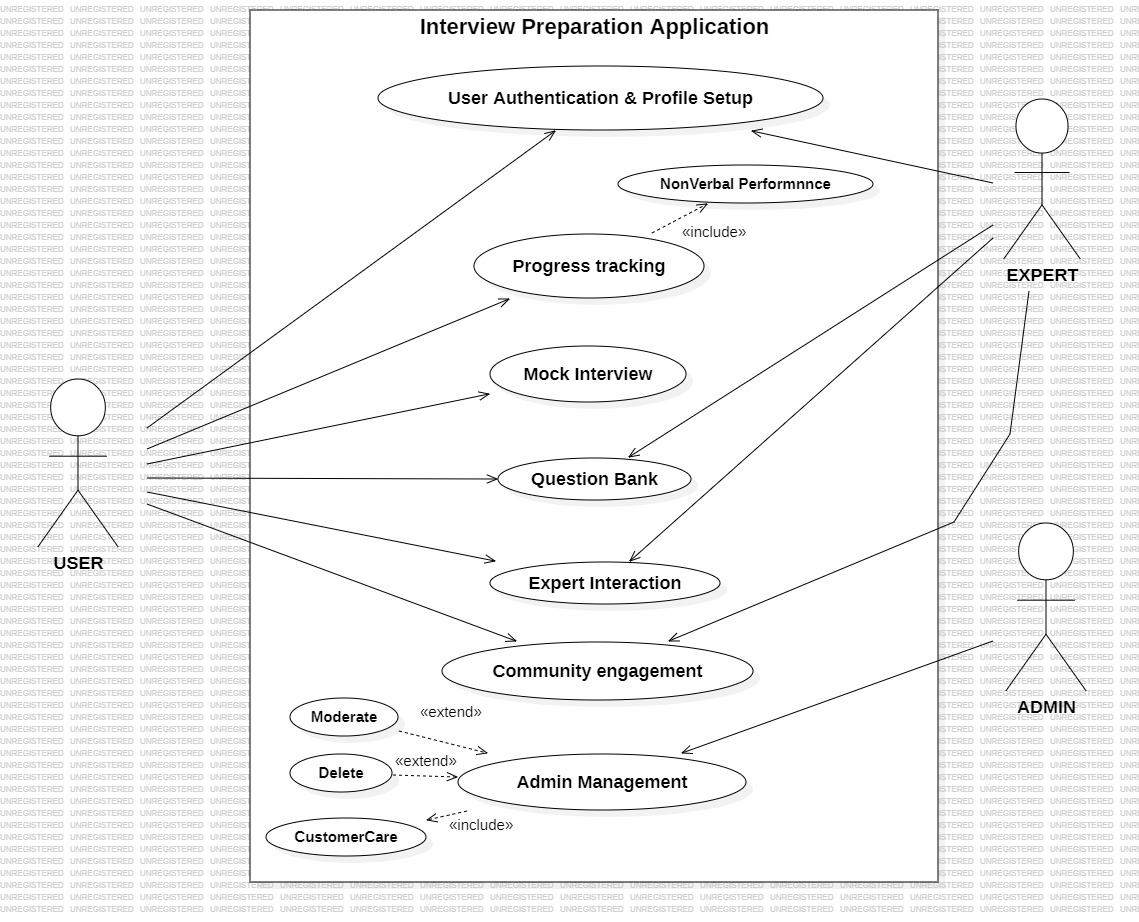


*Figure 2.1: Huru Interface*

1. **Requirement Analysis**

The following parts of the Software Requirements Specification (SRS) report should be included in this chapter.

2. 1. **Use Cases Diagram**

****

*Figure 3.1: Main Use Case Diagram*

* 1. **Detailed Use Case**

*Table 3.1: User Authentication and Profile Setup*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1 |
| **Use Case Name:** | User Authentication and Profile Setup |
| **Actors:** | **Primary Actor:** User, **Secondary Actor:** Expert |
| **Description:** | The user creates a profile and sets up their interview preparation goals. |
| **Trigger:** | The user clicks on the "Signup" button. |
| **Preconditions:** | The user must not be already registered. |
| **Postconditions:** | The user has a valid email address. |
| **Normal Flow:** | 1. The user opens the application and is prompted with the login screen. 2. The user enters their registered email address and password. 3. The application verifies the credentials and grants access to the user's account. 4. The user is directed to their dashboard where they can set up their profile by providing additional information such as their educational background and technical interests. |
| **Alternative Flows:** | None |
| **Exceptions:** | If the user enters an incorrect email or password, the application displays an error message and prompts the user to retry |
| **Business Rules** | The user must provide the correct login credentials that match the information stored in the application's database. |
| **Assumptions:** | The user has a stable internet connection during the authentication process. |

*Table 3.2: Progress Tracking*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-2 |
| **Use Case Name:** | Progress Tracking |
| **Actors:** | User |
| **Description:** | This use case involves tracking the user's progress and performance within the Interview Preparation Application |
| **Trigger:** | The user accesses the progress tracking feature within the application. |
| **Preconditions:** | The user must not be already registered. |
| **Postconditions:** | The user gains insights into their strengths and areas for improvement. |
| **Normal Flow:** | 1. The user navigates to the progress tracking section within the application. 2. The application compiles and displays the user's performance data, including the areas they have mastered and those requiring further attention. 3. The user can review the data and plan their future preparation strategies accordingly. |
| **Alternative Flows:** | None |
| **Exceptions:** | If the application fails to retrieve the user's progress data, it displays an error message and suggests the user try again later. |
| **Business Rules** | The progress tracking feature relies on accurate data collection from the user's interactions within the mock interview. |
| **Assumptions:** | The user has given the mock interview. |

*Table 3.3: Mock Interview*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3 |
| **Use Case Name:** | Mock Interview |
| **Actors:** | User |
| **Description:** | This use case involves the user conducting a mock interview using the Interview Preparation Application to improve their interview skills and gain confidence. |
| **Trigger:** | The user selects the mock interview option from the application's menu. |
| **Preconditions:** | The user must have completed the user authentication and profile setup process. |
| **Postconditions:** | The user receives feedback on their mock interview performance and areas for improvement. |
| **Normal Flow:** | 1. The user navigates to the mock interview section in the application. 2. The user begins the mock interview, which is recorded by the application. 3. The application analyses the user's performance based on facial expressions, body language, and verbal responses. 4. The user receives feedback on their confidence levels and non-verbal communication skills. |
| **Alternative Flows:** | None |
| **Exceptions:** | If the user experiences technical difficulties during the mock interview, the application displays an error message and prompts the user to try again. |
| **Business Rules** | The mock interview feature relies on the user's device camera and microphone functionality. |
| **Assumptions:** | The user has a device with a functional camera and microphone for the mock interview. |

*Table 3.4: Question Bank*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4 |
| **Use Case Name:** | Question Bank |
| **Actors:** | **Primary Actor:** User, **Secondary Actor:** Expert |
| **Description:** | This use case involves users accessing a comprehensive question bank within the Interview Preparation Application to explore various interview questions and prepare effectively. |
| **Trigger:** | The user selects the question bank option from the application's menu. |
| **Preconditions:** | The user must have completed the user authentication and profile setup process. |
| **Postconditions:** | The user gains access to a wide range of interview questions and answers for practice and preparation. |
| **Normal Flow:** | 1. The user navigates to the question bank section in the application. 2. The application displays a categorized list of interview questions based on different software engineering roles and topics. 3. The user selects a specific category or topic to view related interview questions and their suggested answers. |
| **Alternative Flows:** | None |
| **Exceptions:** | If the question bank section experiences technical issues, the application displays an error message and prompts the user to try again later. |
| **Business Rules** | The question bank content is regularly updated and reviewed for accuracy and relevance. |
| **Assumptions:** | The user has a stable internet connection to access the question bank content in the application. |

*Table 3.5: Expert Interaction*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5 |
| **Use Case Name:** | Expert Interaction |
| **Actors:** | **Primary Actor:** User, **Secondary Actor:** Expert |
| **Description:** | This use case involves users interacting with experts who have given interview or have some knowledge about interview within the Interview Preparation Application to seek guidance, advice, and valuable insights for their interview preparation. |
| **Trigger:** | The user selects the expert interaction option from the application's menu. |
| **Preconditions:** | The user must have completed the user authentication and profile setup process. |
| **Postconditions:** | The user gains valuable advice and insights from experienced industry professionals. |
| **Normal Flow:** | 1. The user navigates to the expert interaction section in the application. 2. The user selects a specific expert to interact with and sends a request or message seeking guidance. |
| **Alternative Flows:** | None |
| **Exceptions:** | If an expert is unavailable or unresponsive, the application provides alternative options or prompts the user to try connecting with other available experts. |
| **Business Rules** | The expert interaction feature respects the privacy and availability settings of industry experts within the application. |
| **Assumptions:** | The user sends professional and courteous messages to the industry experts, fostering a respectful and informative interaction. |

*Table 3.6: Community Engagement*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6 |
| **Use Case Name:** | Community Engagement |
| **Actors:** | **Primary Actor:** User, **Secondary Actor:** Expert |
| **Description:** | This use case involves users participating in the community engagement activities provided within the Interview Preparation Application, such as discussion forums, sharing experiences, and providing feedback. |
| **Trigger:** | The user selects the community engagement option from the application's menu |
| **Preconditions:** | The user must have completed the user authentication and profile setup process. |
| **Postconditions:** | The user actively engages in informative discussions, shares experiences, and provides feedback to other users |
| **Normal Flow:** | 1. The user navigates to the community engagement section in the application. 2. The application displays various discussion topics, threads, or forums related to software engineering interviews and preparation strategies. 3. The user actively participates by sharing experiences, providing feedback to other users, and contributing to ongoing discussions. |
| **Alternative Flows:** | None |
| **Exceptions:** | If the community engagement section experiences technical issues, the application displays an error message |
| **Business Rules** | The community engagement feature encourages constructive and respectful interactions among users, fostering a supportive and informative community environment. |
| **Assumptions:** | The user respects the community guidelines and code of conduct, promoting positive and beneficial engagement within the community. |

*Table 3.7: Admin Management*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7 |
| **Use Case Name:** | Admin Management |
| **Actors:** | Admin |
| **Description:** | This use case involves the administrator managing and overseeing the operations and content within the Interview Preparation Application, including user activities, expert interactions, and community engagement. |
| **Trigger:** | The admin accesses the application's admin management portal. |
| **Preconditions:** | The admin accesses the application's admin management portal. |
| **Postconditions:** | The admin successfully monitors and moderate’s user activities, expert interactions, and community engagement. |
| **Normal Flow:** | 1. The admin logs into the admin management portal using the provided credentials. 2. The application provides a comprehensive overview of user activities, expert interactions, and community engagement data. 3. The admin moderate’s user-generated content, resolves disputes, and ensures adherence to community guidelines and standards. |
| **Alternative Flows:** | None |
| **Exceptions:** | If the admin encounters technical issues while accessing the admin management portal, the application displays an error message and prompts the admin to try again later. |
| **Business Rules** | The admin management feature grants the administrator the authority to enforce application guidelines and standards, ensuring a safe and supportive environment for all users. |
| **Assumptions:** | The admin performs their duties diligently and impartially, fostering a secure and professional environment within the application. |

* 1. **Functional Requirements**

**FR#1:** Candidate shall authenticate their identity or create an account and set up their profile.

**FR#2:** Candidate shall be able to track their progress after completing a mock interview.

**FR#3**: Candidate shall perform mock interviews to assess and improve their non-verbal skills.

**FR#4:** Candidate shall have access to a question bank to study and practice interview questions.

**FR#5:** Candidate shall be able to interact with experts and seek guidance on interview preparation.

**FR#6:** Candidate shall be able to engage with the community to learn from others' experiences.

**FR#7:** The system should enable administrators to manage user accounts, monitor content, and ensure a positive user experience.

* 1. **Non-Functional Requirements**

**Usability:**

* **USR-1:** The Interview Preparation Application shall have an intuitive and user-friendly interface to facilitate easy navigation and access to features.
* **USR-2:** Error messages and warnings should be presented in a user-friendly manner to guide users in error recovery without confusion.

**Performance:**

* **PRF-1:** The application should load Android devices within 10 seconds of clicking the application icon.
* **PRF-2:** The average response time for any user action within the application should not exceed 1 second under normal load conditions.
* **PRF-3:** The database should be able to handle at least 1000 concurrent users accessing the application without any significant performance degradation.

1. **Design and Architecture**



6. 1. **System Architecture**

For the Interview Preparation Application, a suitable architecture would be the Model-View-Controller (MVC) architecture. This choice is primarily based on the need for a structured and organized approach to handle the application's complex functionalities. MVC provides clear separation between the application's data, user interface, and business logic, promoting better code maintainability, scalability, and reusability.

A diagram of a process

Description automatically generated

*Figure 4.1: MVC Architecture Diagram*

* 1. **Data Representation**

**JSON Tree Class Diagram for Firebase**

JSON Tree Class Diagram is used for the data representation of the system using.

Firebase. In JSON Tree Class the data is stored in the form of a JSON object. JSON objects typically consist of a key/value pair. In this diagram, the user is accessing objects from the database using these four classes: Admin, Expert, Candidates, Question Bank

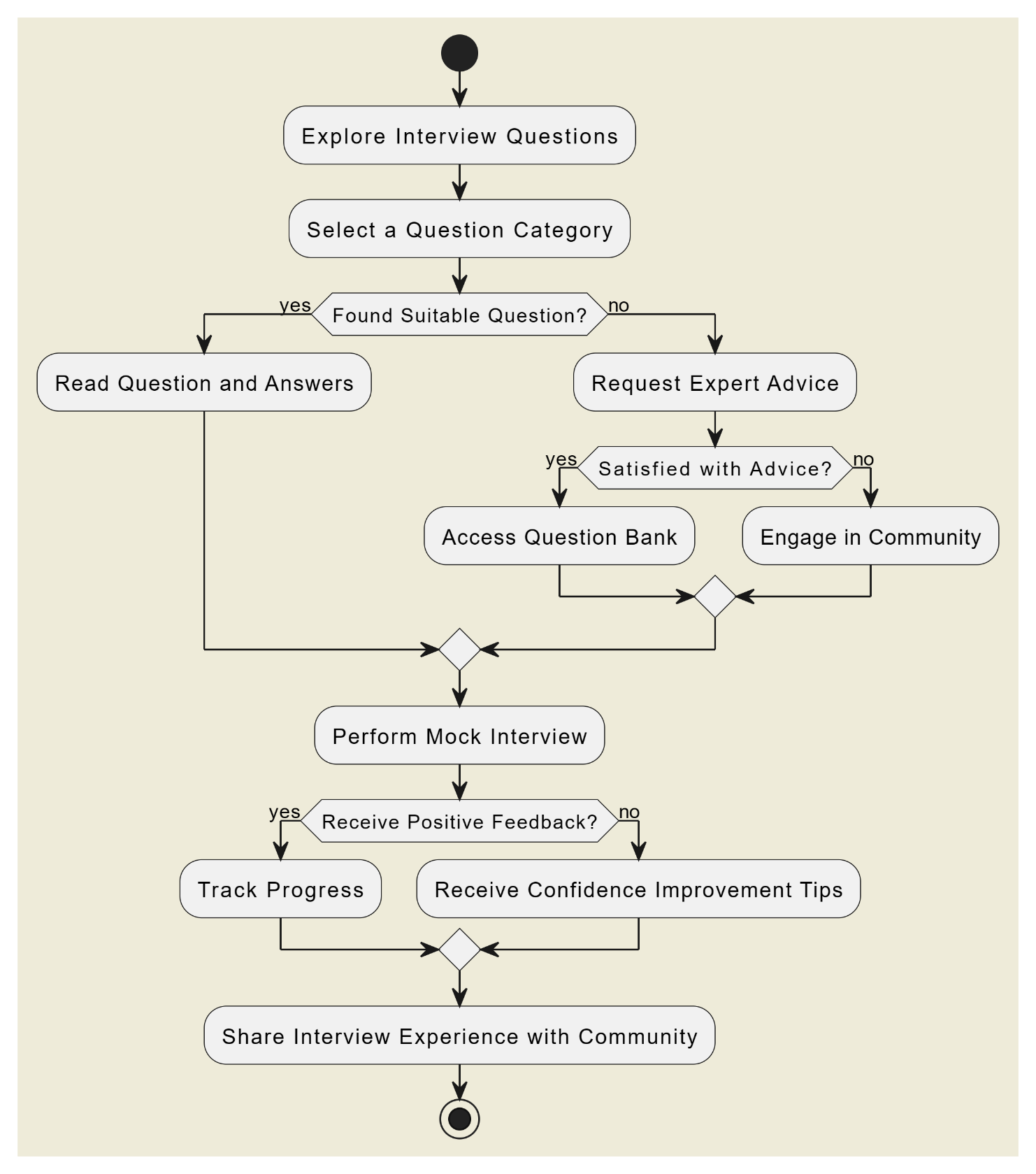
**A screenshot of a computer

Description automatically generated**

*Figure 4.2: JSON Tree Class Diagram*

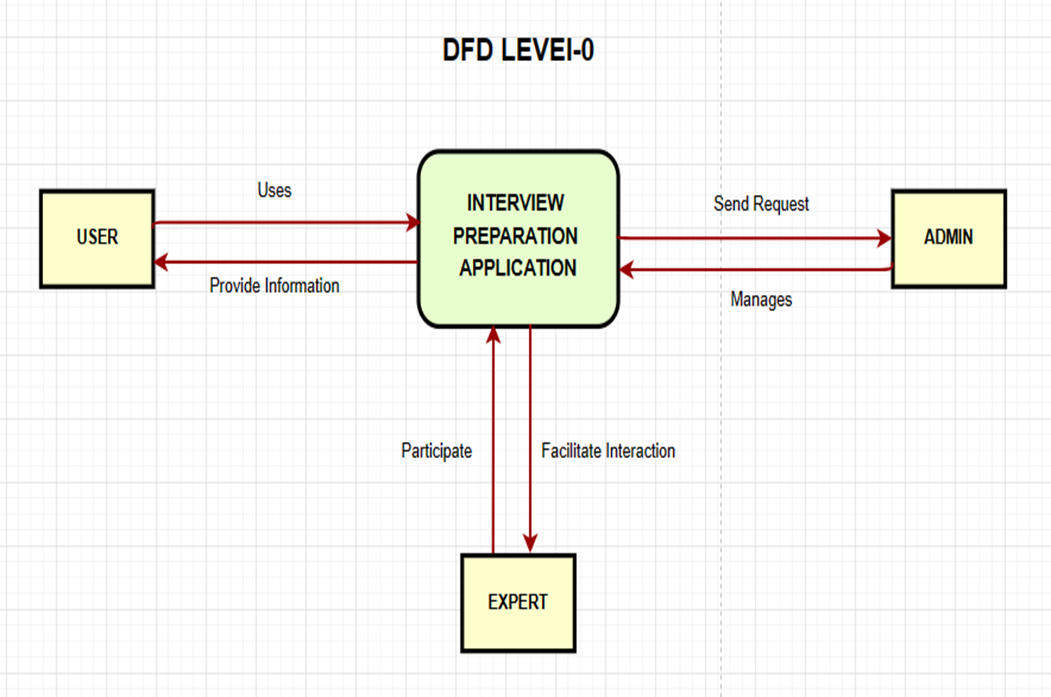
* 1. **Process Flow/Representation**

This process flow diagram is for the whole system which shows the flow that how the user will interact with this system and what he/she can do.



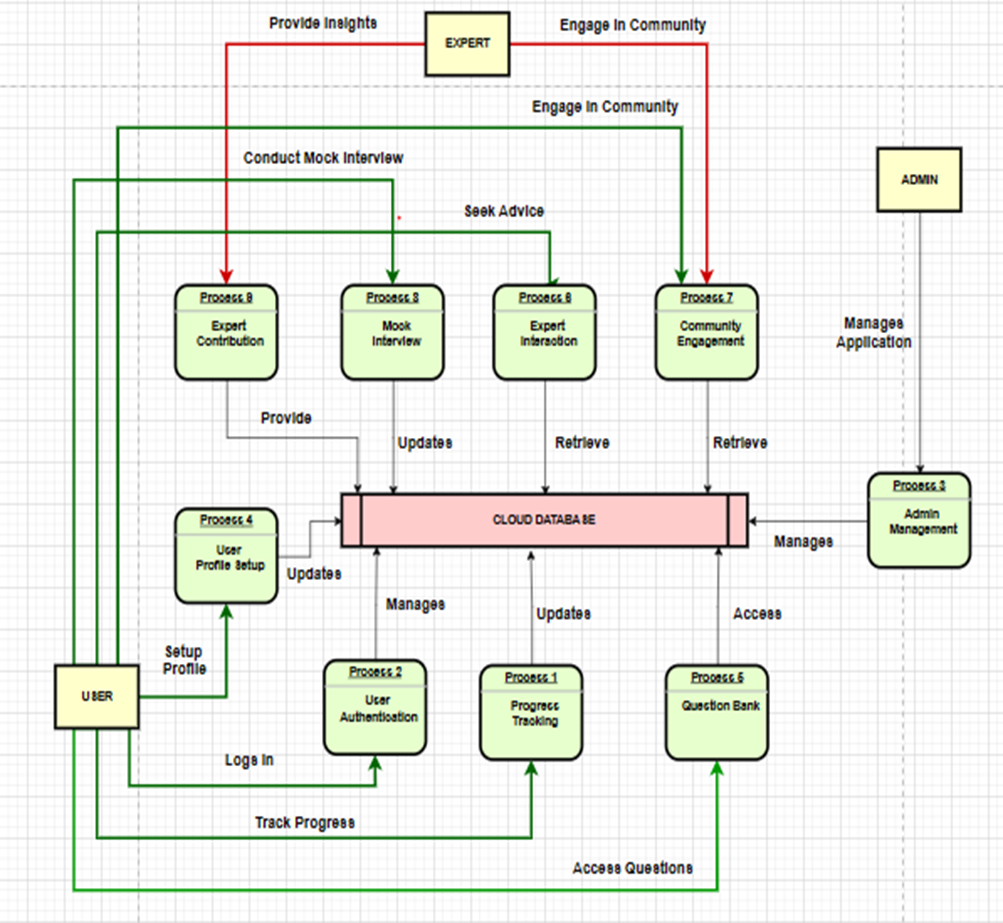
*Figure 4.3: Process Flow/Representation*

* 1. **Design Models**
* **Data Flow Diagram:** 
  + **DFD (Level-0):**

****

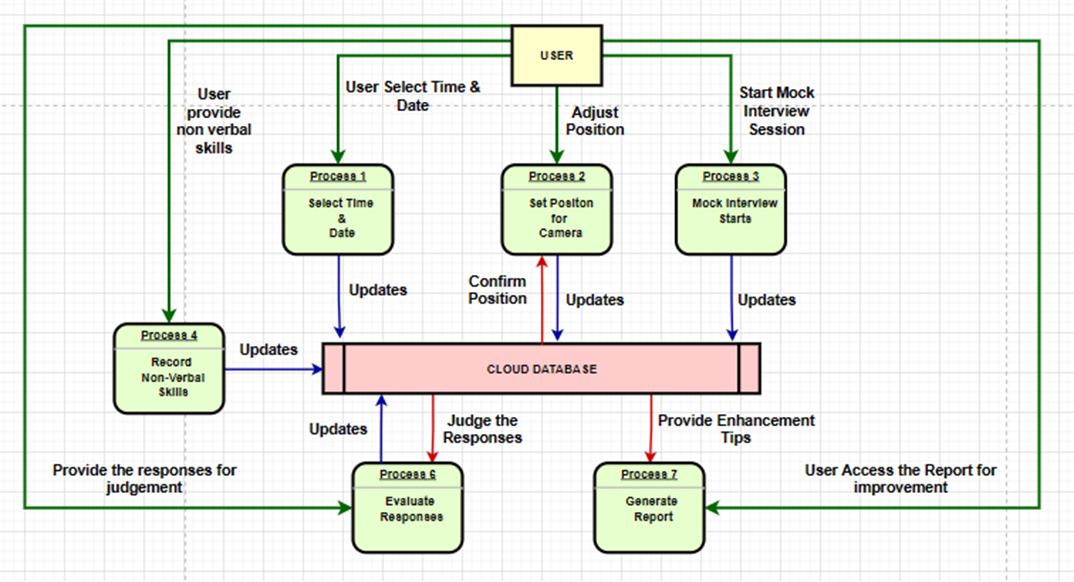
*Figure 4.4: DFD Level-0*

* + **DFD (Level-1):**

****

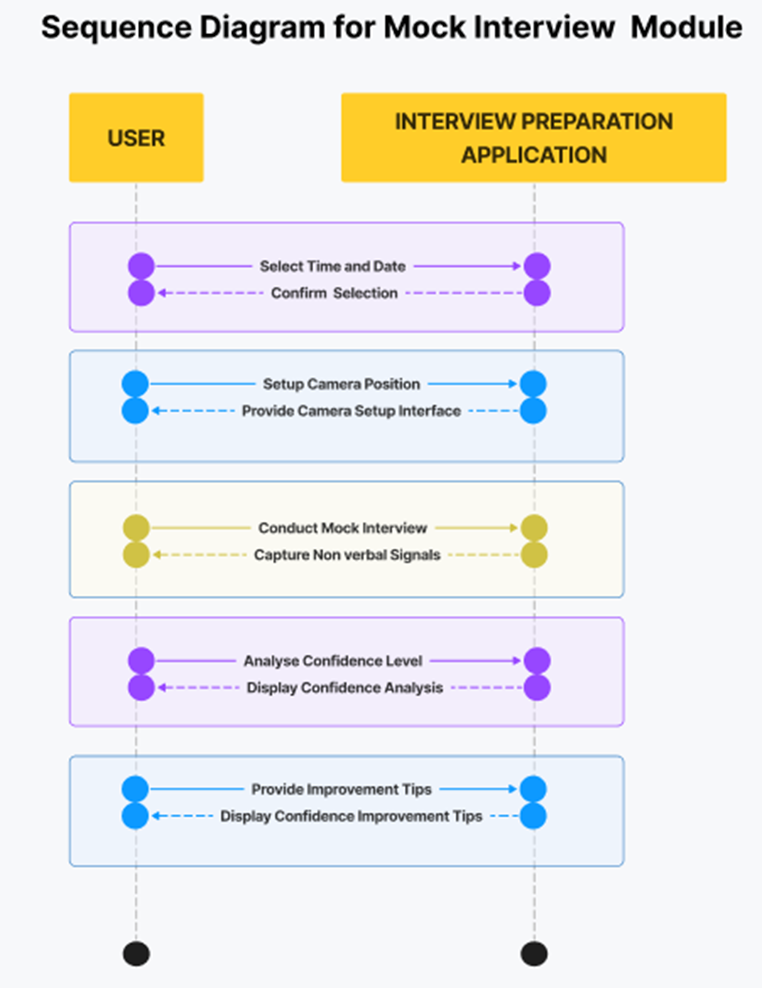
*Figure 4.5: DFD Level-1*

* + **DFD (Level-2) (Mock Interview):**

****

*Figure 4.6: DFD Level-2*

* **Sequence Diagram:**
  + **Sequence Diagram (Mock Interview Module):**

****

*Figure 4.7: Sequence Diagram (Mock Interview Module)*

* + **Sequence Diagram (Expert Module):**

A diagram of a process

Description automatically generated

*Figure 4.8: Sequence Diagram (Expert Module)*

* + **Sequence Diagram (Admin Module):**

A diagram of a system

Description automatically generated

*Figure 4.9: Sequence Diagram (Admin Module)*

* + **Sequence Diagram (User Module):**

A screenshot of a computer screen

Description automatically generated

*Figure 4.10: Sequence Diagram (User Module)*

1. **Implementation**

This chapter will discuss implementation details supported by UML diagrams (if applicable). You will not put your source code here. Any of the following sections may be included based on your project.

1. 1. **Algorithm**

Mention the algorithm(s) used in your project to get the work done with regards to major modules. Provide a pseudocode **OR** a natural language explanation regarding the functioning of main features. Be sure to use the correct syntax and semantics for algorithm representations.

* 1. **External APIs**

Describe the APIs used in the table 5.1.

Table 5.1 shows that

*Table 5.1: Details of APIs used in the project*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of API** | **Description of API** | **Purpose of usage** | **List down the function/class name in which it is used** |
|  |  |  |  |
|  |  |  |  |

* 1. **User Interface**

Details about user interface with descriptions.

1. **Testing and Evaluation**

This chapter may include the following sections. (Students are required to perform the testing both manually and automatedly).

3. 1. **Manual Testing**

This is the sample text

1. * 1. **System testing**

Once the system has been successfully developed, testing has to be performed to ensure that the system working as intended. This is also to check that the system meets the requirements stated earlier. Besides that, system testing will help in finding the errors that may be hidden from the user. There are few types of testing which includes the unit testing, functional testing and integration testing. The testing must be completed before it is being deploy for user to use.

* + 1. **Unit Testing**

Once the system has been successfully developed

* **Unit Testing 1:** Login as FYP Committee as shown in Table 5.1

**Testing Objective:** To ensure the login form is working correctly

*Table 5.1: Login Unit Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Verify user login after click on the ‘Login’ button on login form with correct input data | Username:  L001  Password:  1234 | Successfully log into the main page of the system as FYP Committee member. | Pass |
| 2. |  |  |  |  |

* **Unit Testing 2:** Edit Profile

**Testing Objective:** To ensure the edit profile form is working properly.

*Table 5.2: Edit Profile Unit Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. |  |  |  |  |

* + 1. **Functional Testing**

The functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

* **Functional Testing 1:** Login with different roles as shown in Table 5.3

**Objective**: To ensure that the correct page with the correct navigation bar is loaded.

*Table 5.3: Login Functional Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Login as a ‘FYP Committee’ member. | Username: L001  Password: 1234 | Main page for the FYP Committee member is loaded with the FYP Committee navigation bar | Pass |
| 2. |  |  |  |  |

* + 1. **Integration Testing**

Table 5.4 shows the integration testing

*Table 5.4: Integration Testcase*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1. | Login as “FYP Committee” member | Username: L001  Password: 1234 | Login successful and the FYP Committee page with its navigation bar is loaded and in the view profile page | Pass |
| 2. | Upload student record for Project 1 | - | File successfully uploaded and return to the upload page. Student records are updated. | Pass |
| 3. | View supervising student | - | The list of supervisees shown on the screen. | Pass |

* 1. **Automated Testing:**

This is the sample text

* + 1. **Tools used:**

Table 5.5 shows the

*Table 5.5: Tools used*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool Name** | **Tool Description** | **Applied on [list of related test cases / FR / NFR]** | **Results** |
|  |  |  |  |
|  |  |  |  |

1. **Conclusion and Future Work**

This chapter concludes the project and highlights future work.

1. 1. **Conclusion**
   2. **Future Work**
2. **References**

References to any book, journal paper or website should properly be acknowledged. Please consistently follow the style. The following are few examples of different resources i.e. journal article, book, and website.

* 1. Lyda M.S. Lau, Jayne Curson, Richard Drew, Peter Dew and Christine Leigh, (1999), Use Of VSP Resource Rooms to Support Group Work in a Learning Environment, ACM 99, pp-2. **(Journal paper example)**
  2. Hideyuki Nakanishi, Chikara Yoshida, Toshikazu Nishmora and TuruIshada, (1996), FreeWalk: Supporting Casual Meetings in a Network, pp 308-314 **(paper on web)** http://www.acm.org/pubs/articles/proceedings/cscw/240080/p308-nakanishi.pdf
  3. Ali Behforooz& Frederick J.Hudson, (1996), Software Engineering Fundamentals, Oxford University Press. Chapter 8, pp255-235. **(book reference example)**
  4. Page Author, Page Title, http://www.bt.com/bttj/archive.htm, Last date accessed**. (web site)**