PROJECT TITLE SHOULD BE IN UPPERCase LETTERS & TIMES NEW ROMAN

A PROJECT REPORT

BY

TEAM NO. 54

TEAM MEMBER1 (E23CSE001)

TEAM MEMBER2 (E23CSE002)

A blue and red text on a black background

Description automatically generated TEAM MEMBER3 (E23CSE003)

SUBMITTED TO

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND TECHNOLOGY, BENNETT UNIVERSITY

GREATER NOIDA, 201310, UTTAR PRADESH, INDIA

April 2025

# DECLARATION

I/We hereby declare that the work which is being presented in the report entitled “Project Title”, is an authentic record of my/our own work carried out during the period from JAN, 2023 to April, 2023 at School of Computer Science and Engineering and Technology, Bennett University Greater Noida.

The matters and the results presented in this report has not been submitted by me/us for the award of any other degree elsewhere.

Signature of Candidate

Name of Teammate1

(Enroll. No. E22CSE001)

Name of Teammate2

(Enroll. No. E22CSE002)

Name of Teammate3

(Enroll. No. E22CSE003)

Name of Teammate3

(Enroll. No. E22CSE004)

# ACKNOWLEDGEMENT

I/We would like to take this opportunity to express my/our deepest gratitude to my/our mentor, **Dr. IJKL (**Provide correct name & designation) for guiding, supporting, and helping me/us in every possible way. I/we was/were extremely fortunate to have him as my/our mentor as he provided insightful solutions to problems faced by me/us thus contributing immensely towards the completion of this capstone project. I/We would also like to express my/our deepest gratitude to VC, DEAN, HOD, faculty members and friends who helped me/us in successful completion of this capstone project. Any other name you can mentioned here. (Acknowledgement is your personal view, so you can write it in your way by maintaining integrity of technical report).

Signature of Candidate

Name of Teammate1

(Enroll. No. E17CSE001)

Name of Teammate2

(Enroll. No. E17CSE002)

Name of Teammate3

(Enroll. No. E17CSE003)

Name of Teammate3

(Enroll. No. E17CSE003)

**TABLE OF CONTENTS**

<<Right click on the heading and click update flied. Your heading will be pulled here. Same goes with all the table of content in the subsequent pages>>

[LIST OF TABLES vi](#_Toc85881407)

[LIST OF FIGURES vii](#_Toc85881408)

[LIST OF ABBREVIATIONS viii](#_Toc85881409)

[ABSTRACT ix](#_Toc85881410)

[1. INTRODUCTION 1](#_Toc85881411)

[1.1. Problem Statement 1](#_Toc85881412)

[2. Background Research 2](#_Toc85881413)

[2.1. Proposed System 2](#_Toc85881414)

[2.2. Goals and Objectives 2](#_Toc85881415)

[3. Project Planning 3](#_Toc85881416)

[3.1. Project Lifecycle 3](#_Toc85881417)

[3.2. Project Setup 3](#_Toc85881418)

[3.3. Stakeholders 4](#_Toc85881419)

[3.4. Project Resources 4](#_Toc85881420)

[3.5. Assumptions 5](#_Toc85881421)

[4. Project Tracking 5](#_Toc85881422)

[4.1. Tracking 5](#_Toc85881423)

[4.2. Communication Plan 6](#_Toc85881424)

[4.3. Deliverables 7](#_Toc85881425)

[5. SYSTEM ANALYSIS AND DESIGN 8](#_Toc85881426)

[5.1. Overall Description 8](#_Toc85881427)

[5.2. Users and Roles 8](#_Toc85881428)

[5.3. Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams 9](#_Toc85881429)

[5.3.1. Use Case Diagrams 9](#_Toc85881430)

[5.3.2. Class Diagram 10](#_Toc85881431)

[5.3.3. Activity Diagrams 11](#_Toc85881432)

[5.3.4. Sequence Diagram 12](#_Toc85881433)

[5.3.5. Data Architecture 13](#_Toc85881434)

[6. User Interface 14](#_Toc85881435)

[6.1. UI Description 14](#_Toc85881436)

[6.2. UI Mockup 14](#_Toc85881437)

[7. Algorithms/Pseudo Code 15](#_Toc85881438)

[8. Project Closure 16](#_Toc85881439)

[8.1. Goals / Vision 16](#_Toc85881440)

[8.2. Delivered Solution 16](#_Toc85881441)

[8.3. Remaining Work 16](#_Toc85881442)

[REFERENCES 17](#_Toc85881443)

LIST OF TABLES

Table Page

[Table 1: Goal and Objectives 2](#_Toc20994041)

[Table 2: Sample 2 3](#_Toc20994042)

[Table 3: Sample 3 4](#_Toc20994043)

[Table 4: Sample 4 4](#_Toc20994044)

[Table 5: Sample 4 5](#_Toc20994045)

[Table 6: Sample 6 5](#_Toc20994046)

[Table 7: Regularly Scheduled Meetings 6](#_Toc20994047)

[Table 8: Information To Be Shared Within Our Group 6](#_Toc20994048)

[Table 9: Information To Be Provided To Other Groups 6](#_Toc20994049)

[Table 10: Information Needed From Other Groups 7](#_Toc20994050)

[Table 11: Deliverables 7](#_Toc20994051)

[Table 12: Sample 12 8](#_Toc20994052)

LIST OF FIGURES

Figure Page

[Figure 1: Sample use-case diagram 9](#_Toc20994053)

[Figure 2: sample 2 11](#_Toc20994054)

[Figure 3: sample 3 12](#_Toc20994055)

[Figure 4: Sample 4 13](#_Toc20994056)

[Figure 5: Sample 5 13](#_Toc20994057)

[Figure 6: Sample 6 14](#_Toc20994058)

LIST OF ABBREVIATIONS

Abbreviation Explanation of the Abbreviation

AAA Authentication Authorization and Access Control

CSP Cloud Service Provider

DNS Domain Name System

IAM Identity and Access Management

ABSTRACT

Abstracts must use 250 words to 500 Basic guidelines for the preparation of a technical work for the project report. This document is itself an example of the desired layout (inclusive of this abstract) and can be used as a template. The document contains information regarding desktop publishing format, type sizes, and typefaces. Style rules are provided that explain how to handle equations, units, figures, tables, abbreviations, and acronyms. Sections are also devoted to the preparation of acknowledgments, references, and authors' biographies. The abstract is limited to 500 words and cannot contain equations, figures, tables, or references. It should concisely state what was done, how it was done, principal results, and their significance.

1. INTRODUCTION

<<Provide a brief overview of the current trends and situation around your project>>

Paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text

* Bulleted list items use the “Bullets” style from the styles pane.
  1. Problem Statement

<< Provide a concise statement on the problem that currently exists and is affecting the organization/society/task.

Example: Currently, Microsoft uses Microsoft Dynamics AX as their enterprise resource planning (ERP) solutions for businesses. The latest version of Microsoft Dynamics AX will collect a significant amount of telemetry about the user’s actions while navigating in the application and its thousands of forms.>>

1. Background Research

<<Provide a detailed description of the literature research **(atleast 2 pages)** that you conducted around your project topic (make sure to cite the relevant sources in your text. E.g., research papers/web articles/blogs) and what motivated you, after researching, to work on the project.

Would suggest you to download a tool (Mendeley: <https://www.mendeley.com>) for automated citation and generation of references.>>

* 1. Proposed System

<< Provide a concise statement on the purpose of the project; the problem or opportunity addressed. The explanation should include what you intended to do. Vision – how will the customer’s world improve as a result of this project? When appropriate, tie this into what is currently being considered or has just been completed at the organization

Example: This project aims to extract information about how users use the forms by analyzing the raw telemetry data. By using big data analysis and machine learning techniques we hope to develop predictions into what actions the user will do next. Using these insights, the Microsoft Dynamics AX engineering may be able to optimize the user experience and reduce the number of steps needed to perform desired actions or reach desired forms.>>

* 1. Goals and Objectives

<< State, in quantifiable terms, if possible, the goals and objectives of the project. Goals may be related to product, process, quality, or teamwork. >>

**Example:**

<<to insert caption for tables and figures, click on references on the top menu, under captions section 🡪 click insert caption 🡪 choose label as table>>

Table 1: Goal and Objectives

|  |  |
| --- | --- |
| **#** | **Goal or Objective** |
| 1 | Make the system extensible – future updates like xxx can be done easily |
| 2 | Make the system easy to support – provide good documentation, configuration/build files, administrator’s manual |
| 3 | Make the system very easy to use – users would agree that minimal to no training is needed |
| 4 | Build a prototype that demonstrates the user interface by xx/xx/xx - in order to get early feedback from the customer/users |
| 5 | Have fun working on the project |

1. Project Planning

This section covers the details of the project planning. Selecting the lifecycle of the development, project stakeholders, resources required, assumptions made (if any) are detailed in the sections below.

* 1. Project Lifecycle

<< Describe the lifecycle of the project. You can choose from an existing lifecycle definition or create your own.

Example: The team will use an agile approach. Our team will gather requirements and create a high level development plan at the onset of the project and then implement the gathered requirements over three iterations. The team will follow a SCRUM-like approach with an emphasis on frequent meetings and collaboration.>>

* 1. Project Setup

<< Define some of the basic project decisions that will be used on this project. >>

**Example:**

Table 2: Sample 2

|  |  |
| --- | --- |
| **#** | **Decision Description** |
| 1 | Windows 8, C#, OpenSphere vs. Azure, Trac/SVN vs. Git, etc. |
| 2 | Standards that must be followed (default Capstone coding standard, etc.) |
| 3 | Special access privileges needed, nondisclosure forms, release to open source, etc. |
| 4 | A virtual server image will be set up at NDSU that matches the customer environment (image provided by customer) |

* 1. Stakeholders

<< Identify all stakeholders for this project (groups or individuals that are affected by or are in some way accountable for the outcome of the project – business managers, end users, developers, testers, support people, instructors, etc.) >>

**Example:**

Table 3: Sample 3

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Person A | Sponsor |
| Person B | Mentor |
| Person C | Instructor |
| Person D | Team member |
| Person E | Team member |
| Person F | Team member |

* 1. Project Resources

<< Identify the anticipated resources required for this project. This can include staff members who will work on the project, equipment needed for the project, special software that will need to be acquired, or any other resource necessary for the project. >>

**Example:**

Table 4: Sample 4

|  |  |  |
| --- | --- | --- |
| **Resource** | **Resource Description** | **Quantity** |
| Database Server | A database server provided by the sponsoring company. | 1 |
| Capstone Team | Our team of students who will be the primary developers of the project. | 4 |
| Jim Somebody | The mentor who will be able to provide us with technical assistance. | 1 |
| Mac Workstation | An OS X workstation with X Code for developing the OS X version of the software. | 1 |
| Android Phone | An Android phone to be used as test hardware for the mobile version of the software. | 2 |

* 1. Assumptions

<<State any assumptions upon which the project is based. Assumptions may be related to staffing, resources, tools, and schedules/deadlines. >>

**Example:**

Table 5: Sample 4

|  |  |
| --- | --- |
| **#** | **Assumption** |
| A1 | The capstone team and mentors will be able to meet face to face once a week. |
| A2 | Azure ML will be available for the team to work with as a trial for the first month of the project. |
| A3 | Team members will be able to familiarize themselves with the Azure ML, Azure HDInsights, and R environments |
| A4 | Team will have sufficient time to complete a working model to present by mid-semester |
| A5 | Machine Learning model will be completed in time to test on true big data using HDInsights and Hadoop |
| A6 | The development test data provided will be sufficient to create an accurate prediction of user actions |
| A7 | The models developed will be easily extended to other forms within the time frame |

1. Project Tracking
   1. Tracking

<< Provide information about how the project was tracked and where information was kept. This should include information such as what type of source control was being used and how it can be accessed, any bug-tracking system that was used for the project and where it can be accessed, what type of regressing testing suite was used and where it can be accessed, and any similar information that provides details on the project’s status, etc. >>

**Example:**

Table 6: Sample 6

|  |  |  |
| --- | --- | --- |
| **Information** | **Description** | **Link** |
| Code Storage | Project code will be stored in SVN repository. | Link |
| Bug Tracking | Bug tracking will be done with Trac. | Link |
| Project Documents and Assignments | Weekly reports, specification and design documents, etc. will be stored in our SVN repository. | Link |
| Continuous Integration | Continuous integration will be done with Jenkins. | Link |
| Regression Testing | Regression testing will use JUnit unit tests and Jenkins. | Link |

* 1. Communication Plan

<< Identify all communications you will provide to other groups and all communications you need to receive from other groups. Share this information with affected groups. Verify that all stakeholders are included. >>

Table 7: Regularly Scheduled Meetings

|  |  |  |
| --- | --- | --- |
| Meeting Type | Frequency/Schedule | Who Attends |
| Conference Call/Skype | Weekly | Project team and mentor |
| Team Meeting | Weekly | Project team |
| Short Meeting | Weekly in class | Project team |
| Sprint Planning Meeting | Start of each sprint | Project team and mentor |
| Sprint Retrospective Meeting | End of each sprint | Project team |
| Sprint Review Meeting | End of each sprint | Project team, ***mentor, and sponsor*** |

Table 8: Information To Be Shared Within Our Group

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Project team | Task assignments & General scrum information | Weekly | Team meetings, listing in Project Specification. |

Table 9: Information To Be Provided To Other Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Sponsor and mentor | Final deliverables | At completion of project | Project specification doc., code, Power Point presentation |
| Sponsor and mentor | Weekly report | Weekly | Email and Trac site access |
| Sponsor and mentor | Project baselines ***(optional)*** | At the end of each sprint | Onsite customer demo, access to repository |

Table 10: Information Needed From Other Groups

|  |  |  |  |
| --- | --- | --- | --- |
| Who? | What Information? | When? | How? |
| Sponsor and mentor | Requirement changes | Start of each sprint | Conference call or meeting with sponsor and mentor. |
| Nathan Olson | Availability of test server | Start of second sprint | Email |

* 1. Deliverables

<< Identify the major deliverables that this project is expected to produce. Assume the deliverables apply to all features or stories listed above unless indicated otherwise. Deliverables may include prototypes, documentation, software, etc. >>

Table 11: Deliverables

|  |  |
| --- | --- |
| **#** | **Deliverable** |
| 1 | Study results ***(if any)*** |
| 2 | Code |
| 3 | Test and test results |
| 4 | Build process documents***(if any)*** |
| 5 | Install process documents***(if any)*** |
| 6 | Administrator or user manual***(if any)*** |
| 7 | Postmortem document |
| 8 | Final report (final PowerPoint presentation, 3 minute video, and final sprint) |

1. SYSTEM ANALYSIS AND DESIGN

This section describes in detail about the design part of the system.

* 1. Overall Description

<< Provide a more detailed, two to three paragraph description of the project. This description may include more technical details to describe the purpose of the project.

Example: This project is an attempt to apply data science and machine learning techniques to telemetry data from Microsoft ERP products in an attempt to anticipate user actions based on previous navigation and controls to create more efficient application navigation. Using Event Tracer for Windows, test teams are able to record millions of data points with information regarding actions that users have taken, forms they have navigated through, and the time data for each of those events. By combing the data and arranging it by session ID and time we can create a sort of roadmap of each user’s actions in time order from the moment they start the application.

First this data must be cleaned and sorted using R statistical software. Then graph visualizations of the data (initially pertaining to the navigation path through the application) will be generated to allow us to view the dominant paths to specific modules and forms in the product, and give us an idea of where to start with machine learning. The statistics and visualization using R will be compiled into an R package to allow for easy documentation, and extensible use with various datasets. The visualization and statistics functions will provide basic insight into the data for any Program Manager, or someone not experienced in data science. >>

* 1. Users and Roles

<< Provide a list and description of the different types of users or roles within the system. This may include different classes of users, such as administrator, instructor, student, etc. This list may also include autonomous agents that interact with the system as well. These may include users (or personas) that are used as part of any user stories produced for the project. >>

**Example:**

Table 12: Sample 12

|  |  |
| --- | --- |
| **User** | **Description** |
| Developer | A capstone team member or mentor who is tasked with managing the test data, creating initial machine learning models, and ultimately generating a firm process for applying these techniques to future user data. This is used for sub-stories and task needed to fulfill the true end user use cases. |
| Microsoft Program Manager | A manager at Microsoft who is working on developing the ERP application who will be making design decisions based on the data analysis. |
| Dynamics AX User | An end user of the Microsoft ERP product who will be generating the data used and reaping the potential efficiency benefits from the data analysis when designing the application. |

* 1. Design diagrams/Architecture/ UML diagrams/ Flow Charts/ E-R diagrams

<<Provide all the design diagrams that were created during the design phase of your project. Those who love coding may use a tool <https://liveuml.com/> to create UML diagrams. You may refer <https://plantuml.com/> for commands/syntax of various diagrams. Some sample may be checked below:

* Use case: <https://liveuml.com/view/6024c067492d8164f7df07a3>
* Class: <https://liveuml.com/view/6024c3e8492d8164f7df07a4>

Some visual examples are in sections below:>>

* + 1. Product Backlog Items

<<All features/services/functionalities are known as **Product Backlog Items (PBIs)**. When some features are picked from PBIs to develop in a SPRINT of two weeks, then it is known as **Sprint Backlog Items (SBIs).** The features are then written as user stories, acceptance criteria, and tasks in a SPRINT.>>

Enlist all of your functionalities/requirements as user-stories in the format below:

As **a <user type>** I want **<actions>** So that **<achieved goal>.**

**Examples:**

* "As Max, I want to invite my friends, so we can enjoy this service together."
* "As Sascha, I want to organize my work, so I can feel more in control."
* "As a manager, I want to be able to understand my colleagues' progress, so I can better report our success and failures."
* "As a fitness enthusiast, I want to track my workouts and progress in an app so that I can monitor my improvements."
* "As a frequent shopper, I want to save my payment information so that I can check out quickly on future purchases."
* "As an online student, I want to receive notifications for upcoming classes so that I can easily manage my schedule."
  + 1. Architecture Diagram

<< Provide the architecture diagram (may be a pattern selected) for your product>>.

* + 1. Use Case Diagram

<< Provide any use-case diagrams that are being used as part of the project. Uniquely label each use case so that if necessary, it is easy to reference from other parts of the document. >>

**Example:** Restaurant system

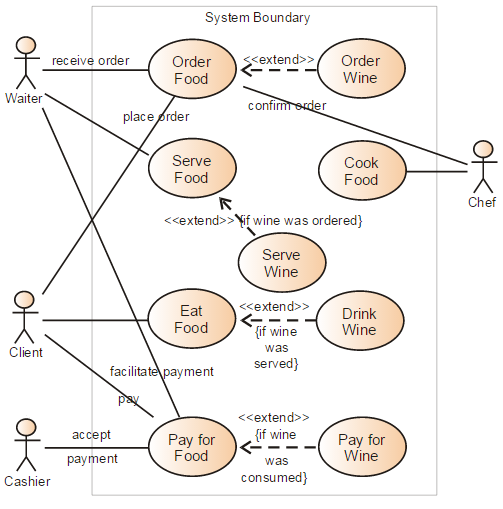


Figure 1: Sample use-case diagram

<<to insert caption for tables and figures, click on references on the top menu, under captions section 🡪 click insert caption 🡪 choose label as table

If you use a table, figure, or non-text item that is not your original design, you must cite the original source of the item. You may use an in-text citation in the text of the title or caption of the item, or you may include the citation as a footnote. Refer to the style manual of your discipline for more information about citations of non-text items.>>

* + 1. Class Diagram

<< Include a class diagram for all classes to be designed. Optionally include major data elements of those classes and important methods and functions that will be used by other classes.>>

**Example:** Online Photo Collection

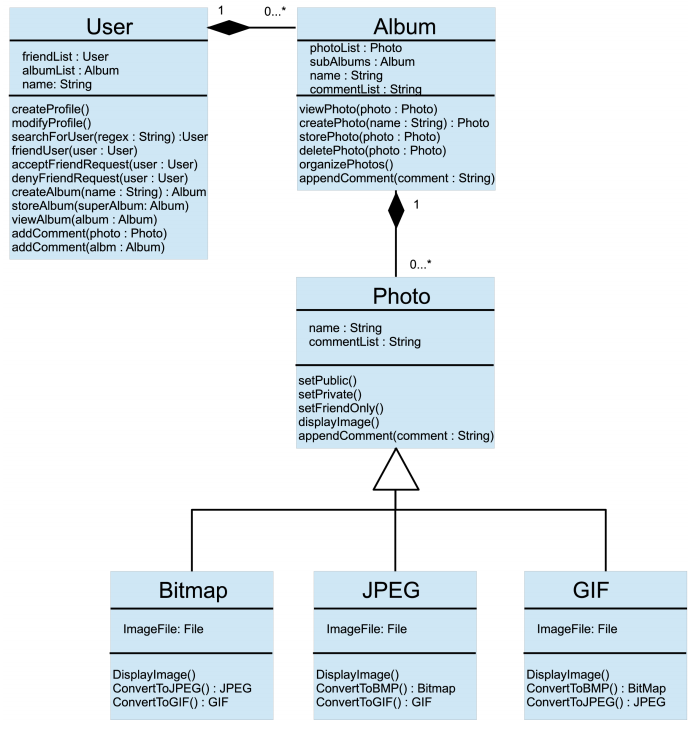


Figure 2: sample 2

* + 1. Activity Diagrams

<< Include activity diagrams for important workflows in the program. At least one diagram should be included for the main workflow in the program. Optionally include labels that indicate which component is responsible for that part of the activity. Activity diagrams for components which perform complex tasks should also be included. >>

**Example**: View friend’s photo album

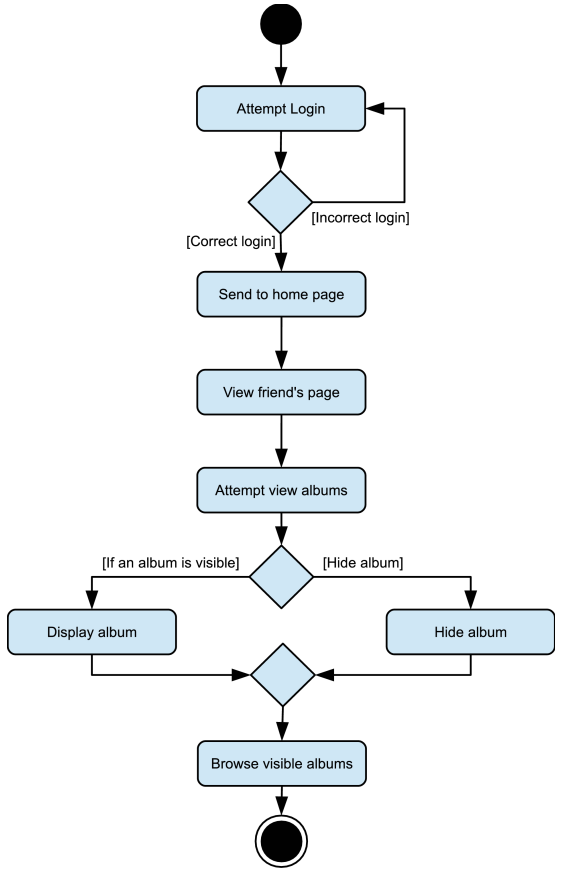


Figure 3: sample 3

* + 1. Sequence Diagram

<< Include sequence diagrams for important functionality of the program to indicate control flow. These diagrams should include classes found in the class diagram and use the methods for those classes to show the interaction between them. >>

Example: Create new album

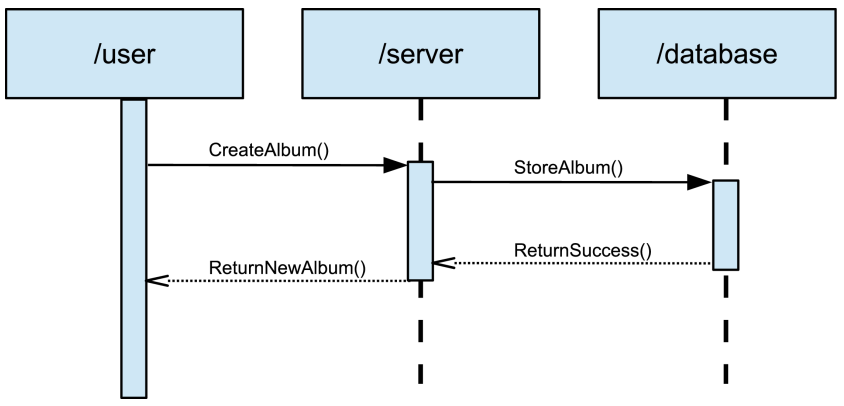


Figure 4: Sample 4

* + 1. Data Architecture

<< Include any information or diagrams that provide details about databases, xml configuration files, or other data structures that are a part of the system. If a very specific format is required, it may be worthwhile to provide a more robust description or a detailed design such as a database schema. >>

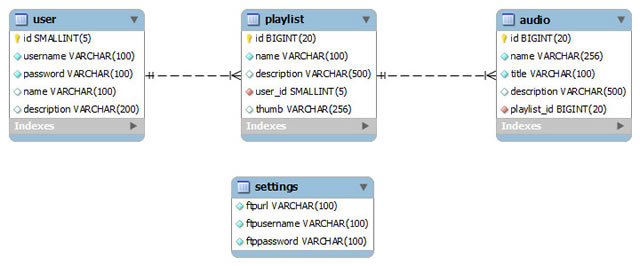


Figure 5: Sample 5

1. User Interface
   1. UI Description

<< Provide a brief description of the UI that will be used in this program and how users will interact with the program.

Example: We are creating a research project using R and Azure ML. R is a statistical programming language that uses the R console, which will be the primary means of interacting with our code. This uses standard console UI, and it is not in the scope of our project to create a UI on top of that. The Azure ML machine learning model has its own built-in drag and drop UI for the different code modules we create, so there is no need to generate any sort of AI for that portion of the project either. >>

* 1. UI Mockup

<< Attach the screenshot of user interface. This can be a simple drawing that demonstrates key parts of the user interface.>>

Example:

Graphical user interface, application, Word

Description automatically generated

Figure 6: Sample 6

1. Algorithms/Pseudo Code OF CORE FUNCTIONALITY

<<Include the proposed algorithm/pseudo code.>>

1. Project Closure

This section elucidates the overall lookup at the project and some of the future works that may enhance the solution.

* 1. Goals / Vision

<< Provide an update to the vision statement that was originally stated in the Project Initiation document.

Example: Our original goals for this project were to take telemetry data from Dynamics AX, analyze and visualize that data, then create machine learning models in Azure ML to predict user navigation based upon previous actions. Through the course of the project, these goals were altered so that the primary goal became creating a well-documented, extensible R language package that facilitated cleaning and importing telemetry data, and contained a variety of useful analysis and visualization functions to make the raw data more understandable. >>

* 1. Delivered Solution

<< Provide a high-level description of what was planned and what is being delivered.

Example: Our solution delivered primarily consisted of a fully documented, fully featured R language package that contained the functions for importing telemetry data, cleaning, separating, and isolating that data, then performing a variety of analysis, statistics, and visualizations on that data. This R package has standard R documentation, a full suite of unit tests, and an integrated manual and help documentation to allow anyone with a basic familiarity with the R environment to utilize our functions. >>

* 1. Remaining Work

<< Provide a short summary of what should be done next, ways of further improving the project, or any additional recommendations.

Example: We created a prototype of a web-based UI for our R package using Shiny in an attempt to make the functionality more accessible for those unfamiliar with R. There are a few minor functions implemented, and a basic UI setup, but a good deal of work would be required to flesh out this web app with all of the functions currently available in the R package, should Microsoft choose to pursue this UI overlay further. >>

REFERENCES

<<Would suggest you to download a tool (Mendeley: <https://www.mendeley.com>) for automated citation and generation of references>>

1. T. Dillon, C. Wu and E. Chang, “Cloud Computing: Issues and Challenges”, 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 27-33, 2010.

2. H. Casanova, et al. Heuristic for scheduling parameters sweep applicationsin grid environments, in: Proceedings of the 9th Heterogeneous Computing Workshop, HCW, pp. 349–363. 2009.

3. Kwok, Yu-Kwong, and Ishfaq Ahmad. "Static scheduling algorithms for allocating directed task graphs to multiprocessors." ACM Computing Surveys (CSUR) 31, Vol. 4, pp. 406-471, 1999.

4. A. Mutz, R. Wolski, Efficient auction-based grid reservation using dynamic programming, in: IEEE/ACM Int’l Conf. on Super Computing, SC 2007.

5. M. Mezmaz, N. Melab, Y. Kessaci, Y. C. Lee, E. G. Talbi, A. Y. Zomaya and D. Tuyttens, “Parallel Bi-Objective Hybrid Metaheuristic for Energy-Aware Scheduling for Cloud Computing Systems”, Journal of Parallel Distributed Computing, Elsevier, Vol. 71, pp. 1497-1508, 2011.

6. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype and Reality for Delivering Computing as the 5th Utility”, Future Generation Computer Systems, Elsevier, Vol. 25, pp. 599-616, 2009.

7. Basements and crawl spaces. Retrieved from http://www.hud.gov/ offices/hsg/sfh/ref/sfhp1-25.cfm (Access in June 2020).