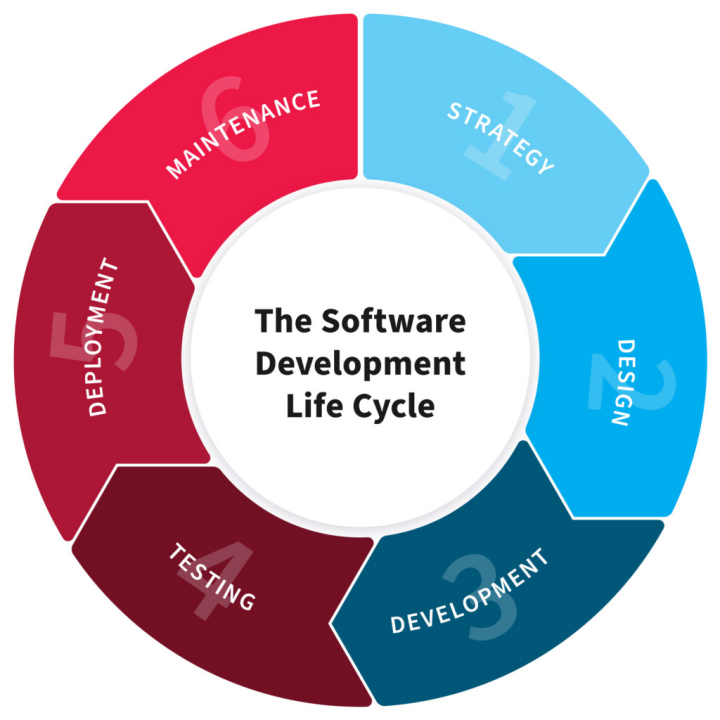
**Format to report your advanced JAVA Programming projects**



**NB:** EVERY STUDENTS HAS TO PRODUCE THIS DOC

Name: NDAYISHIMIYE Reconfort Daniel

RegNo: 221009302

ClassNo: 73

**PROJECT NAME: TODO-APP**

**What is todo-app:** is an application that helps users to create, manage, and track tasks or activities that need to be completed. These apps include features such as the ability to create task lists, set reminders, and mark tasks as complete. They can be used for personal or professional tasks, and are often available for a variety of platforms, such as web, mobile, and desktop.

**Why todo-app:**

There are several reasons why user might use a to-do app:

1. Organization: A to-do app allows users to organize their tasks and activities in one central location, making it easy to see what needs to be done and prioritize tasks based on importance or deadline.
2. Productivity: By using a to-do app, users can better manage their time and stay on top of their responsibilities, which can lead to increased productivity and a sense of accomplishment.
3. Reminders: Many to-do apps include the ability to set reminders, which can help users stay on track and ensure that important tasks are not forgotten.
4. Collaboration: Some to-do apps allow users to share and collaborate on tasks with others, which can be useful for teams or families.
5. Accessibility: With the increasing popularity of mobile devices, many to-do apps are now available as mobile apps, making it easy for users to access their task lists and update them from anywhere.
6. Customization: Many to-do apps offer a variety of customization options, such as different views, color-coding, and more.
7. **STRATEGY**

To create a to-do app in Java, we just used the following strategy:

1. Identify the requirements of the app and create a plan
2. Design the user interface of the app
3. Create a database to store my to-do items with MySQL
4. Implement the functionality of the app using Java.
5. Test the app thoroughly to ensure that it is working
6. Finally, package and distribute the app to users.

**2. PLANNING**

1. We defined scope and goals of our app.
2. We Created a wireframe or a mockup of the app's user interface. This will help you visualise the layout and functionality of the app, and make any necessary adjustments before moving on to the next step.
3. Design and develop the app's backend. This includes creating the database schema, and writing the code for the server-side logic.
4. Develop the front-end of the app. This includes writing the code for the app's user interface, as well as any necessary client-side logic.
5. Test the app thoroughly. This includes running unit tests and functional tests to ensure that the app is working as expected.
6. Deploy the app. This includes hosting the app on a server and making it available to users.
7. Continuously monitor and maintain the app. This includes fixing any bugs that are discovered, and adding new features as needed.

**3. DESIGN**

Through this stage we refers to the overall look and feel, as well as the layout and functionality, of our to-do app. This includes the visual elements such as

* Colors,
* Typography,
* and icons,

as well as the placement of buttons and other interactive elements. The design is user-friendly, visually appealing, and easy to navigate.

In terms of functionality, our app allow users to add, edit, and delete to-do items, as well as provide options to set reminders, and prioritize tasks. The app should also provide a way for users to view their to-do items, such as through a list or calendar view.

**4. DEVELOPMENT**

* We developed the system backend and frontend parts using java programming language by JSP and Servlet
* For frontend parts, we used JSP Framework o create forms, buttons, labels as well as a user interface as a whole.
* We used MySQL as a database management system to hold backend data.
* We create interactions between user interface components such as forms and buttons using Servlet java programming language syntaxes.
* We used com.mysql.jdbc\_5.1.5.jar as a library handling MYSQL connection with the system.

**5. TESTING**

We tested our application with

1. Unit testing: This involves writing individual test cases for specific methods or classes in the application.
2. Integration testing: This involves testing the interactions between different components of the application, such as between the front end and back end.
3. Functional testing: This involves testing the application as a whole, to ensure that it works as intended and meets the requirements.
4. Performance testing: This involves testing the application's performance, such as its response time and memory usage, under different loads and conditions.
5. Manual testing: This involves manually testing the application by going through various scenarios and user flows and checking for any bugs or issues.

**6. DEPLOYMENT**

1. Installing MYSQL as a database management system.
2. Download and configurecom.mysql.jdbc\_5.1.5.jar as a library handling MYSQL connection.
3. o Use portable storage devices to transfer projects from the development computer to any library’s librarian computer.
4. Run the project file and start using the system.

**7. MAINTENANCE**

In our app maintenance we include tasks such as

1. Monitoring and addressing any issues or bugs reported by users
2. Continuously improving and updating the app's user interface and functionality
3. Adding new features or integrations as needed
4. Backing up user data to ensure data is not lost in case of issue
5. Regularly testing the app to ensure it is functioning correctly
6. Addressing any compliance and regulatory issues that may arise.