**Architecture Document**

**1. Introduction**

**1.1 Purpose**

The software architecture document provides a comprehensive overview of the architecture of the software system. It serves as a communication medium between the software architect and other project team members regarding architecturally significant decisions which have been made on the project.

**1.2 Scope**

This Software Architecture Document applies to the Gantt Tech Auto, the Gantt diagram tool, which will be developed by a team that consists of 5 university students.

**1.3 Definitions, Acronyms and Abbreviations**

* Task list: The basic information of task, including task id, task name, starting time, deadline, people in charge.
* Work breakdown structure(WVS): Organizes the team’s work into manageable sections.
* Gantt diagram: There is a timeline above the block, and it shows the order of the task and the bar chart of the length of task.
* Milestone: Mark an importance date in the project, for instance, design complete, phase complete.
* Project: A set of tasks, which can also be represented with a gantt diagram.

**1.4 References**

* Gantter: <https://www.gantter.com/Product/>
* Teamgantt: <https://www.teamgantt.com/>
* Office 365 security whitepaper: <https://www.microsoft.com/en-us/download/details.aspx?id=26552>
* The Benefits of Using a Gantt Chart in Project Management: <https://www.teamgantt.com/blog/10-benefits-of-using-a-gantt-chart-of-your-next-project>

**1.5 Overview**

this document is consist of a few sections below:

* Architectural Representation, briefly describes how many architectures of different views is in this document.
* Architectural Goals and Constraints, describes the software requirements and objectives that have some significant impact on the architecture.
* Use-Case View, lists use cases or scenarios from the use-case model.
* Logical View, describes the architecturally significant parts of the design model.
* Process View, describes the system’s decomposition into processes.
* Implementation View, describes the overall structure of the implementation model.
* Data View, describes the persistent data storage perspective of the system.
* Size and Performance, describes the major dimensioning characteristics of the software that impact the architecture.
* Quality, describes how the software architecture contributes to all capabilities of the system.

**2. Architectural Representation**

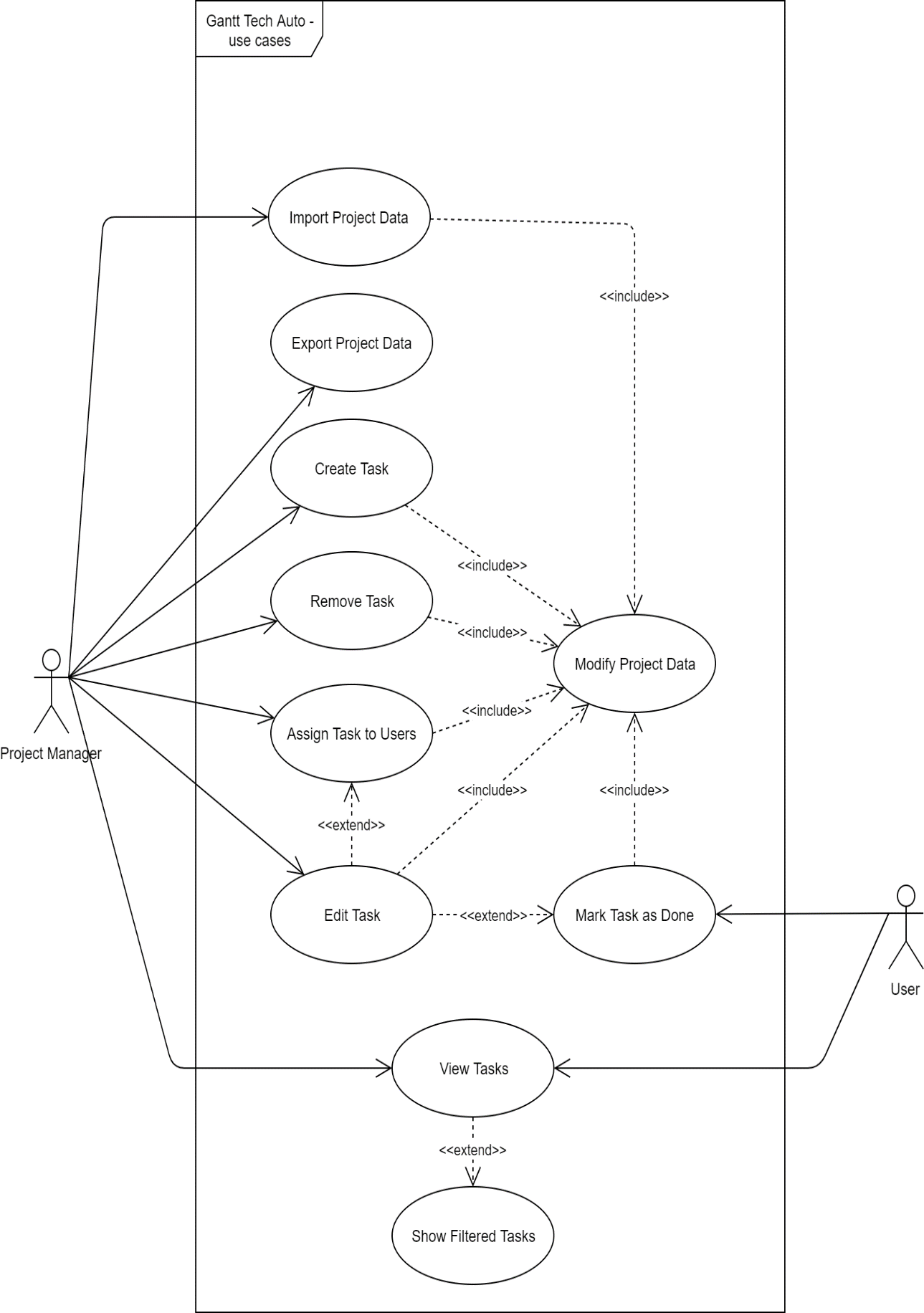
This document presents the architecture as a series of views; use case view, logical view, process view, deployment view and implementation view . These are views on an underlying Unified Modeling Language (UML) model developed using unity.

**3. Architectural Goals and Constraints**

* Goals: The goal of this architecture is to facilitate gantt that is easy to extend and add additional features. All the aspects of the system are created with modularity in mind to help achieve this goal. By having each system functioning as independently as possible, changes made in one area will not have a ripple effect across the system.
* Constraints:
  + Security Constraints: The system should support optional encryption of all user data and activity logs. The encryption keys will be stored on external USB Drive or crypto-card devices that can be stored separately when the device is not in use.
  + Hardware Constraints:  
    function on a system with 2GB of RAM  
    require at most 16GB of disk space

**4. Use-Case View**

The following diagrams depict the use cases in the system:



4.1 Use-Case Realizations

* Project Manager: The leader who is in charge of formulating all tasks related to the project and assigning tasks to other collaboraters.
  + create/remove task, edit task(also assign task to user), only project manager can modify project data
* User: Any other person involved in the project who is responsible for certain tasks
  + user only can view tasks and mark done on task

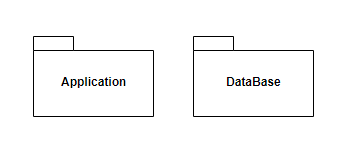
**5. Logical View**

**5.1 Overview**

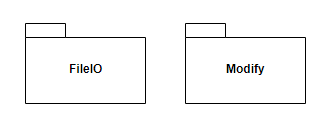
The logical view of the Gantt Tech Auto System is comprised of ? main packages:

* **DataBase**
  + Stored all the information and relationship of the task in a gantt graph.
* **Application**
  + Contains classes for major processing functionality within the system.

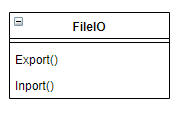
**5.2 Architecturally Significant Design Packages**



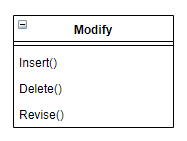
**Application Package**



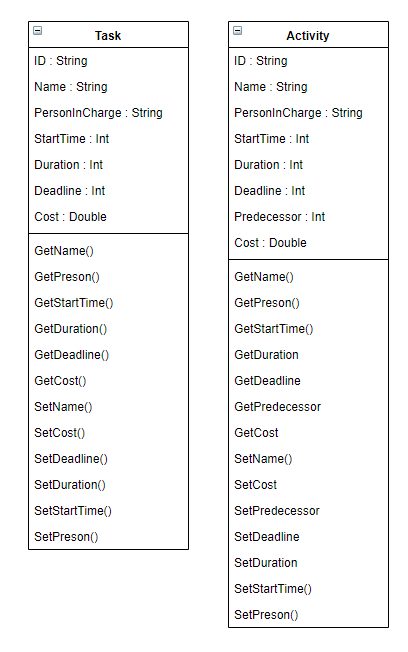
**FileIO**



**Modify**



**DataBase Packag**



**8. Implementation View**

**8.1 Overview**

On user interface, user can access and process the software with supported system.

| **Implementation View Layers** |
| --- |
| User interface |
| User interface management access and process |
| System support (OS, database) |
| Core business logic/application functionality System utilities |

**8.2 Layers**

**10. Size and Performance**

The software architecture supports the key sizing and timing requirements, as stipulated in the Supplementary Specification:

* The system shall provide access to the database with no more than a 10 second latency.
* The system must be able to complete 80% of all transactions within 1 minutes.
* The client portion shall require less than 20 MB disk space and 10 MB RAM.
* Our gantt software are used on pc, so only 1 user using app in a time.
* Our gantt software are used on pc, so local file access only, network is unnecessary.

**11. Quality**

The software architecture supports the quality requirements, as stipulated in the Supplementary Specification:

* The desktop user-interface shall be Windows 10 compliant.
* The app shall be available every times been opened. There shall be no more than 2% down time.
* Upgrades to the PC client portion of app shall be done automatically over the internet. This feature enables users to have easy access to system upgrades.