CSCD01 Deliverable #1

Team 21: *the BugWriters*

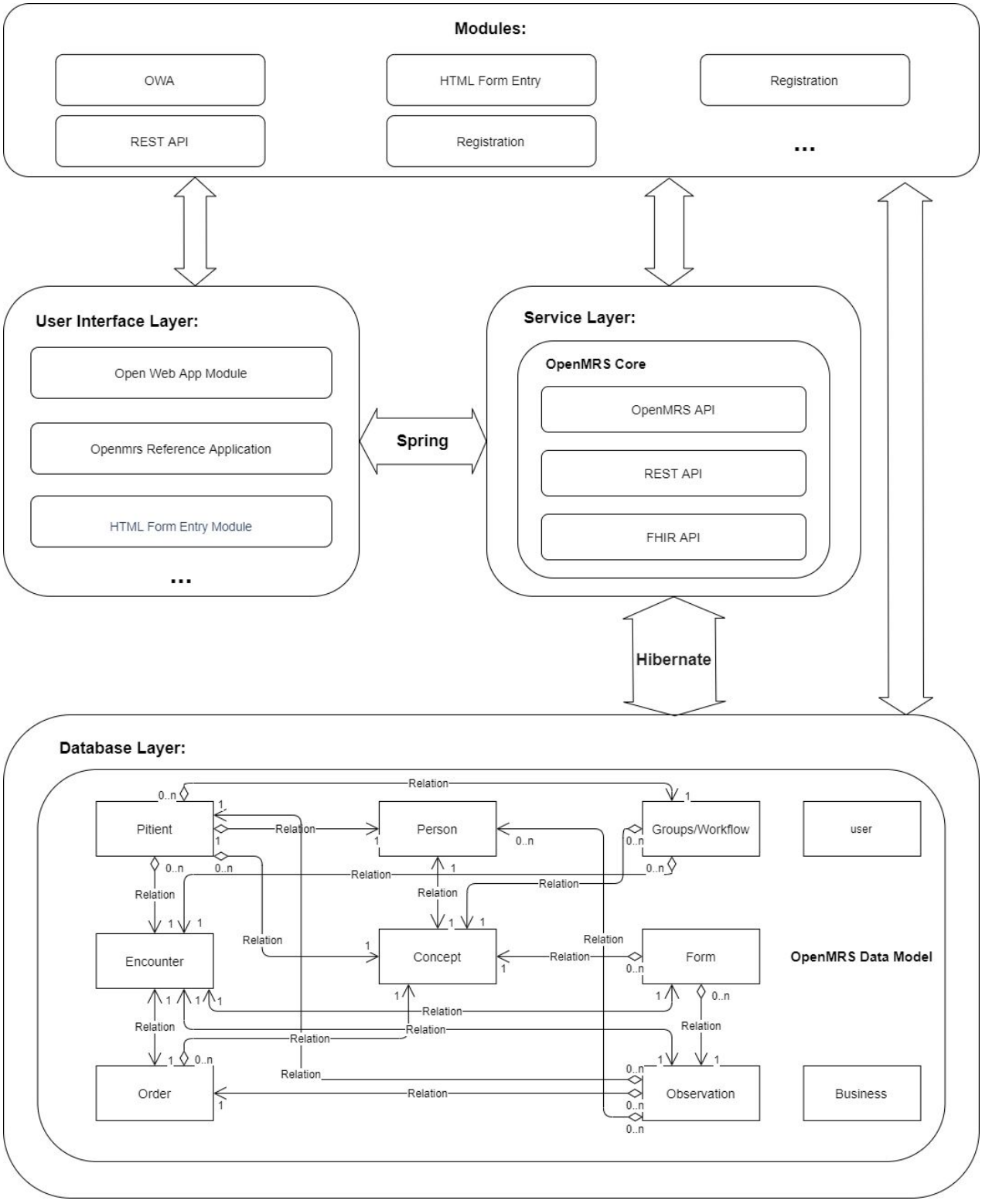
Chengrong Zhang, Zhongyang Xia, Tan Jiaxin, Zongye Yang,

Xingyuan Zhu

February 2020

Contents

0.1 OpenMRS High Level Architecture



Three main layers make the source code of OpenMRS:

1. The user interface layer
2. The service layer
3. The database layer

0.1.1 The User Interface Layer

The user interface layer is responsible for handling different web pages of the system as well as the welcome page, user login page, forum page, etc. The User Interface layer of OpenMRS is built upon Spring MVC, Direct Web Remoting (DWR), JSP and JavaScript.

0.1.2 The Service Layer

The Service layer is responsible for managing the business logic of the application. It is built around the Spring framework. The OpenMRS API has methods for all of the primary functions, such as adding/updating a patient, encounter, observation, etc. The OpenMRS service layer classes make extensive use of the Spring framework for several tasks, including the following:

1. Spring Aspect-Oriented Programming (AOP) is used to provide separate cross-cutting functions (e.g. authentication, logging).
2. Spring Dependency Injection (DI) is used to provide dependencies between components.
3. Spring is used to manage transactions between service layer classes.

0.1.3 The Database Layer

The Data Access layer is responsible for storing the actual data model and its changes. The following are the central databases of the layer at the current version(v2.9.0):

1. Patient
2. Group/Workflow
3. Encounter
4. Observation
5. Person
6. Concept
7. User
8. Form
9. Order
10. Business

0.1.4 Module Architecture

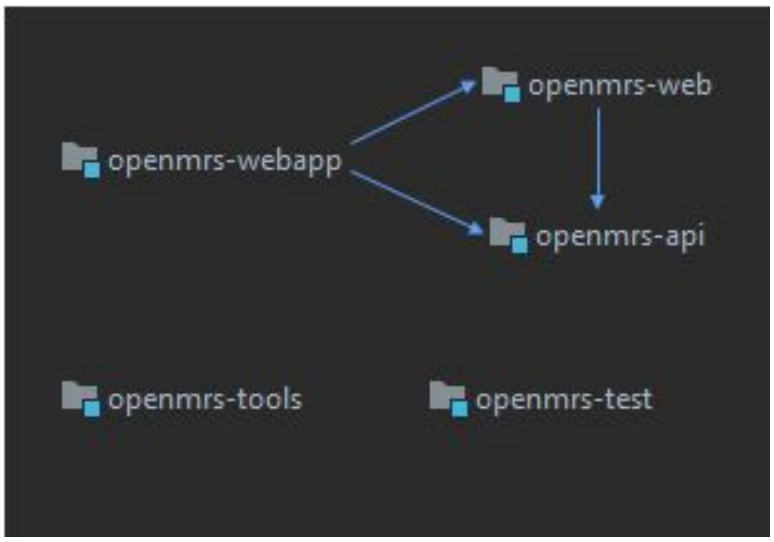
The advantage of OpenMRS is its module framework allows developers to import their customized modules into the software easily. The structure now contains more than 100 different modules, including the Atlas module, Billing module, Api module, REST module, etc.

0.1.5 System Improvement

OpenMRS implements Hibernate DAO to access the database from the service layer since it is easy to use and has a lower risk of data loss compared to sleep mode. But as we all know, Hibernate is not an excellent choice to deal with complex databases because it doesn't allow the insertion of multiple objects in the database using a single query like JDBC. Thus for an extensive system like OpenMRS, Hibernate may perform poorly when numerous data are updating at the same time. Therefore for the improvement, we suggest that OpenMRS use MyBatis to handle communication between service and database.

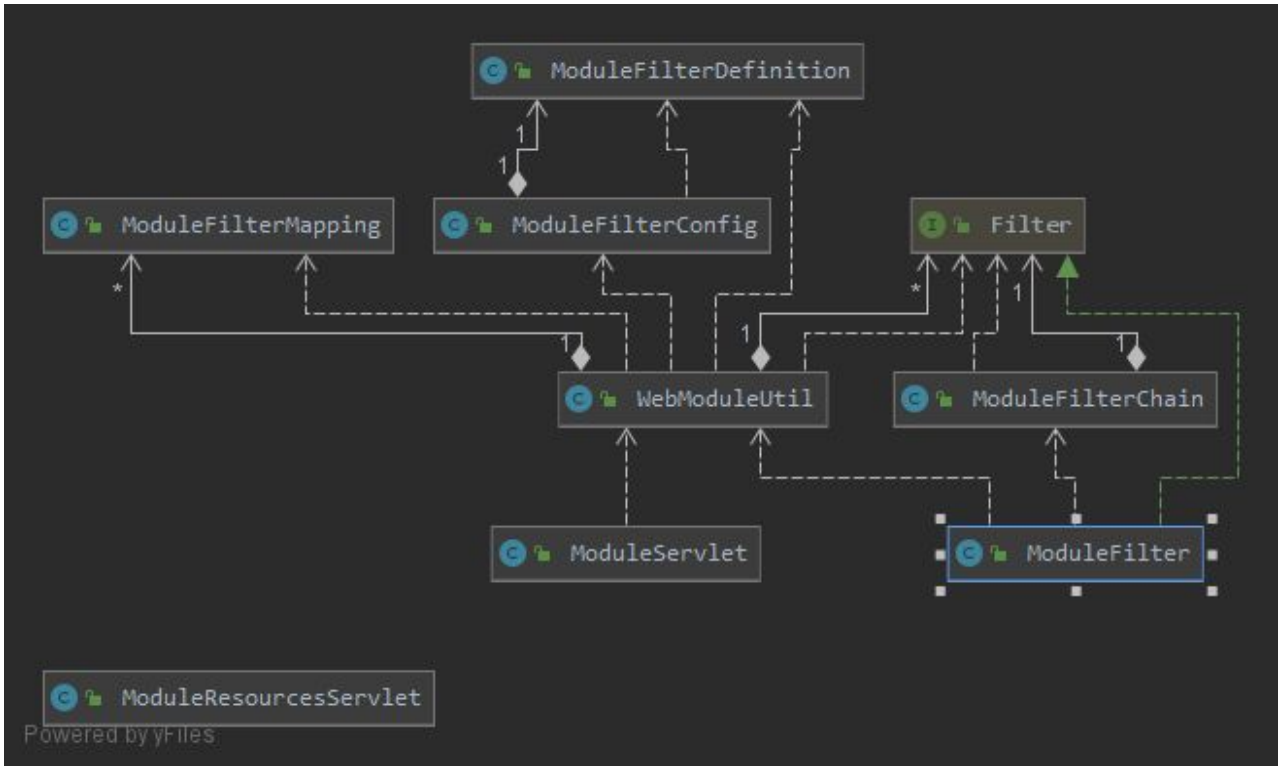
0.2 OpenMRS-core Architecture

0.2.1 Project Modules Diagram

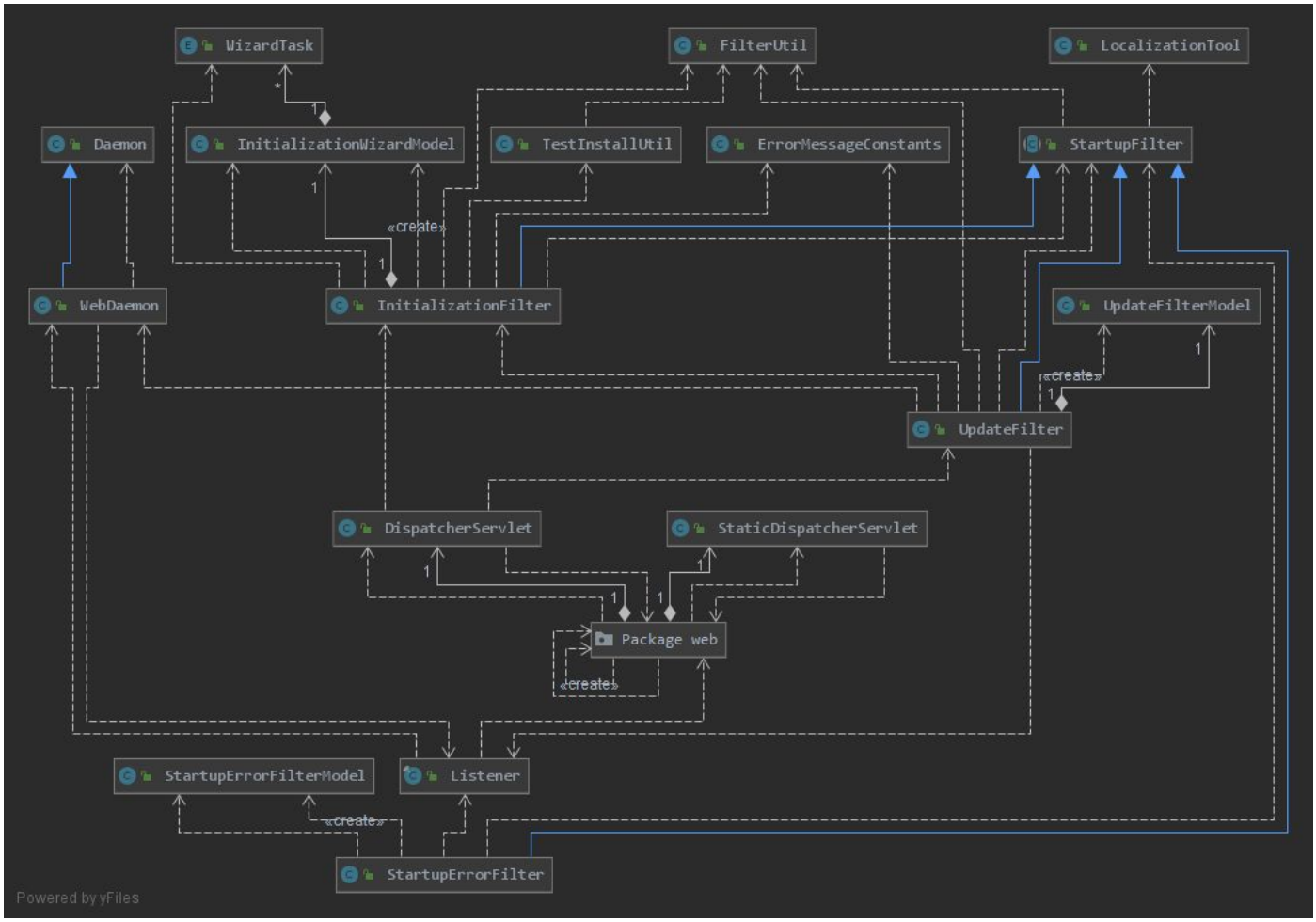


The openmrs-core contains 5 modules. As shown by the diagram, the webapp uses the web and api modules to build the war file for the application.

0.2.2 web/ Folder UML Diagram



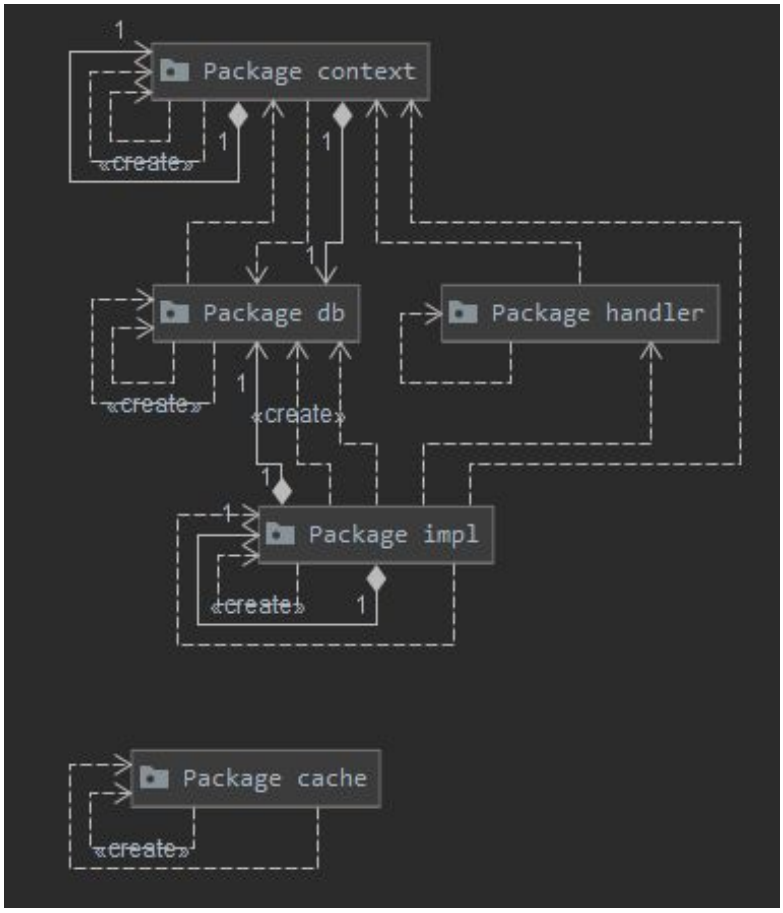
Class diagram for the modular architecture



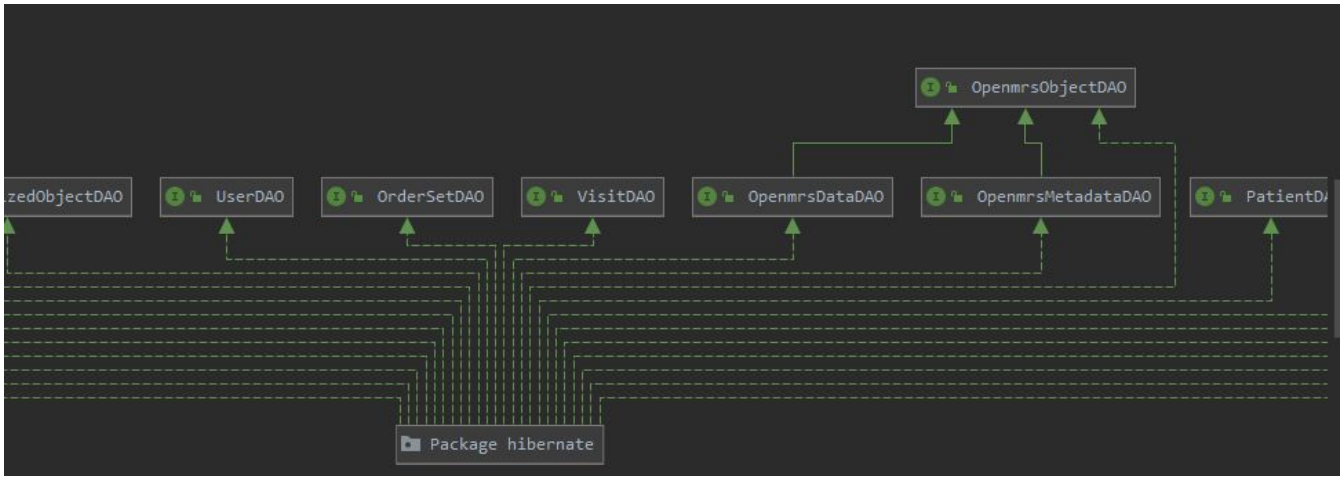
Class diagram for the servlets and web service. 

is represented in the modules class diagram above.

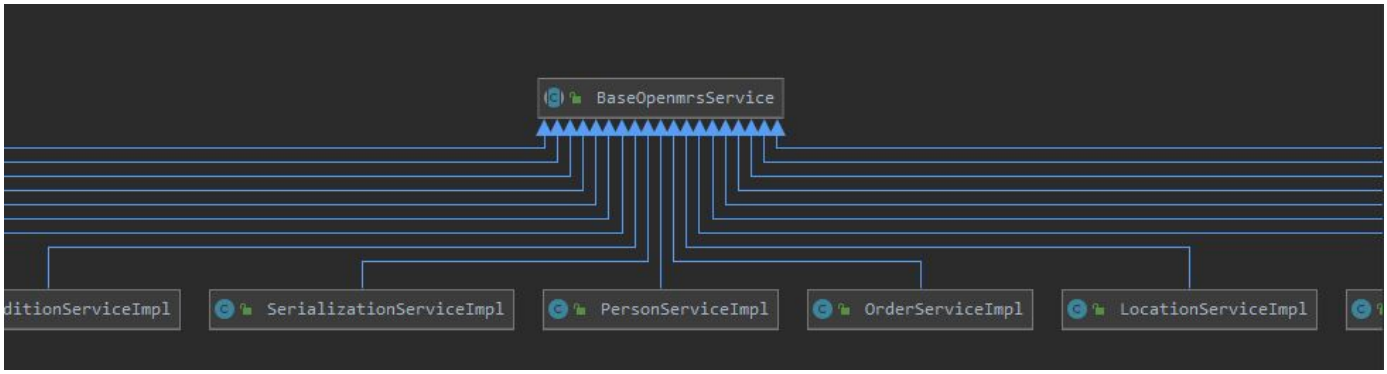
0.2.3 api/ Folder UML Diagram



Java packages diagram for the API service



Part of the UML for the db package, Hibernate is used with data access objects



A small part of the impl package class diagram. Here the implementations of different services are based on the abstract class BaseOpenmrsService.

0.2.4 Overview

Core API is the backbone of OpenMRS. On a higher level, the OpenMRS project utilizes the Spring framework extensively and applied the MVC design pattern. The program contains service and DAO to abstract the database operations and separate some operations that will not interact with the database. Hibernate is used to communicate with underlying MySQL databases. In general, most people use it as a J2EE web application.

OpenMRS uses a “module” system, so that new modules can be easily added to the system for new functionalities or alter existing ones.

OpenMRS contains a modern web app. The jQuery framework is used for the web app, while old versions still use JSP. The web app is connected with the backend using RESTful API calls.

0.3 Software Development Process

Our group chose the *Kanban* development process in the next deliverable. Kanban is a lean method to manage and improve work across human systems. It means to maximize the speed and quality of output, which is suitable for large size program development. The following are the main steps to implementing Kanban process:

0.3.1 Visualization of workflows

We will have a Kanban board using Trello and update the tasks we have to do using cards on the board. The board will have columns for to-do tasks, tasks in progress, tasks being tested, and completed tasks. Each card represents a task and will be placed in one of these columns. These cards contain the description of the task, who it is assigned to, and details for implementation. This way, we can easily visualize the workflow.

0.3.2 Limit work in progress (WIP)

We will set a WIP limit, which restricts the number of tasks we can work on at the same time. This is to improve our focus on a few tasks and to identify bottlenecks. This also prevents an overload of responsibilities.

1. Make policies explicit
2. Manage flow
3. Implement feedback loops

We will have weekly in-person meetings and communicate only daily via WeChat. This is to keep everyone updated on the work in progress and obstacles we are facing.

4. Improve collaboratively, evolve experimentally

The key reason we choose Kanban is its incomparable flexibility. Compared to other software development processes, the Kanban process is more flexible since it does not

require software engineers to have exactly the same skill level, which is closer to our scenario. Also, if anything happens in the middle of the development process, e.g., a group member drops the course - which is quite common and very likely to happen, the Kanban process allows us to handle the changes quickly. What's more, the *feedback loop* process allows us to integrate feedback from all levels. Last but not least, its signature visualization process, the *Kanban*, will enable us to manage the work in a very intuitive way quickly. Therefore, we believe the Kanban development process is an excellent fit for our project and would be very beneficial to us.