SRS: GitGroup

# 4.REQUIREMENTS

## User requirements

### Functional requirements

UR-1: User shall be able to create a new project and new team in GitGroup.

UR-2: User shall be able to manage GitHub repositories and team members like add or remove repositories and members.

UR-3: User shall be able to use Kanban board to organize issues by dragging issue cards among several columns representing different stage of the development processes.

UR-4: User shall be able to analysis the development duration, quality and the team work situation of the project by using a variety of agile charts and analytics.

also manage tasks like suggesting a new idea or tracking a bug.

UR-5: User shall be able to organize an online meeting group chat, schedule and record the conference within the team or the global scope.

### Non-Function requirements

UR-6: The application shall be able to make GitHub easy to use, especially friendly for the person who has little development experience such as product manager.

UR-7: The application shall be quickly restored to operational status after a failure occurs.

UR-8: The app shall be reliable to uses with no downtime.

UR-9: The application shall be able to provide maximum security against malicious attack.

## System requirement

### User Interfaces (Functional Requirements)

UI-1.1: Developers can create a new project and make a team with other developers.

UI-1.2: Developers can remove a project and collaborators of the project after checking if the project is empty.

UI-2.1: Developers can manage their team by inviting to or removing collaborators from their projects.

UI-2.2: Developers can manage their repositories by adding or removing repositories of their projects.

UI-3.1: Developers can classify issues by different stages of development process via putting them in different columns of Kanban board.

UI-3.2: Developers can add or remove a Kanban from their Kanban board.

UI-3.3: Developers can customize their own stage by editing column name except Done column.

UI-3.4: Developers can close issues by dragging it to the Done column, which will automatically close issues in the GitHub.

UI-4.1: Project manager can track and communicate the progress of their projects by burn down and burn up charts.

UI-4.2: Project manager can measure how much work a team can used in eXtreme Programming and Scrum from throughput chart.

UI-4.3: Project manager can analysis the velocity of project going from velocity chart.

UI-5.1: Team leader can organize an online meeting group chat within their team.

UI-5.2: Developers can organize an online meeting group chat within the global.

UI-5.3: Developers can receive a meeting notification from team leader.

UI-5.4: Developers can set an alarm for notifying the coming events on a conference schedule form.

UI-5.5: Developers shall be able to record content of meeting on a meeting notebook.

### Back-end Interfaces(Functional Requirements)

BE-1.3:

### Non-Functional Requirements

NF-6.1: The user shall be able to use all the app functions without any kind of training. The average number of questions call about how to use app shall not exceed 10 per day.

NF-6.2: The app shall be available for any kind of mobile device and PC.

NF-7.1:

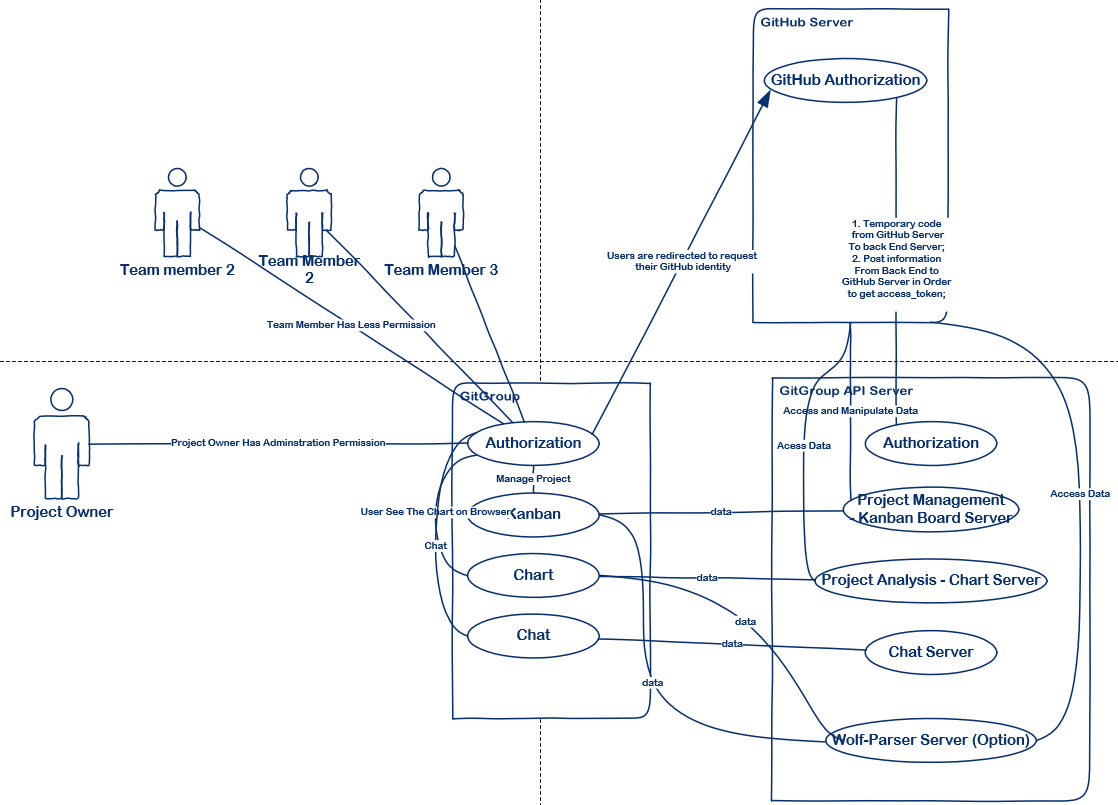
NF-8.1: The app shall be available to all users during whole day (Mon-Sun, 00:00~00:00)

NF-9.1: The app shall not expose contact information to other users.

NF-9.2: The app should minimize the amount of personally identifying information (PII) that it collects.

# 5. SYSTEM MODELS

## Use Case Diagram



User are redirected to request their GitHub identity when they sign in the GitGroup application and then get authorization from GitHub server.

After getting user identity, user can see several columns of Kanban board showing project process status of which data can be toke from Project Management Kanban Board Server and Wolf-Parser Server.

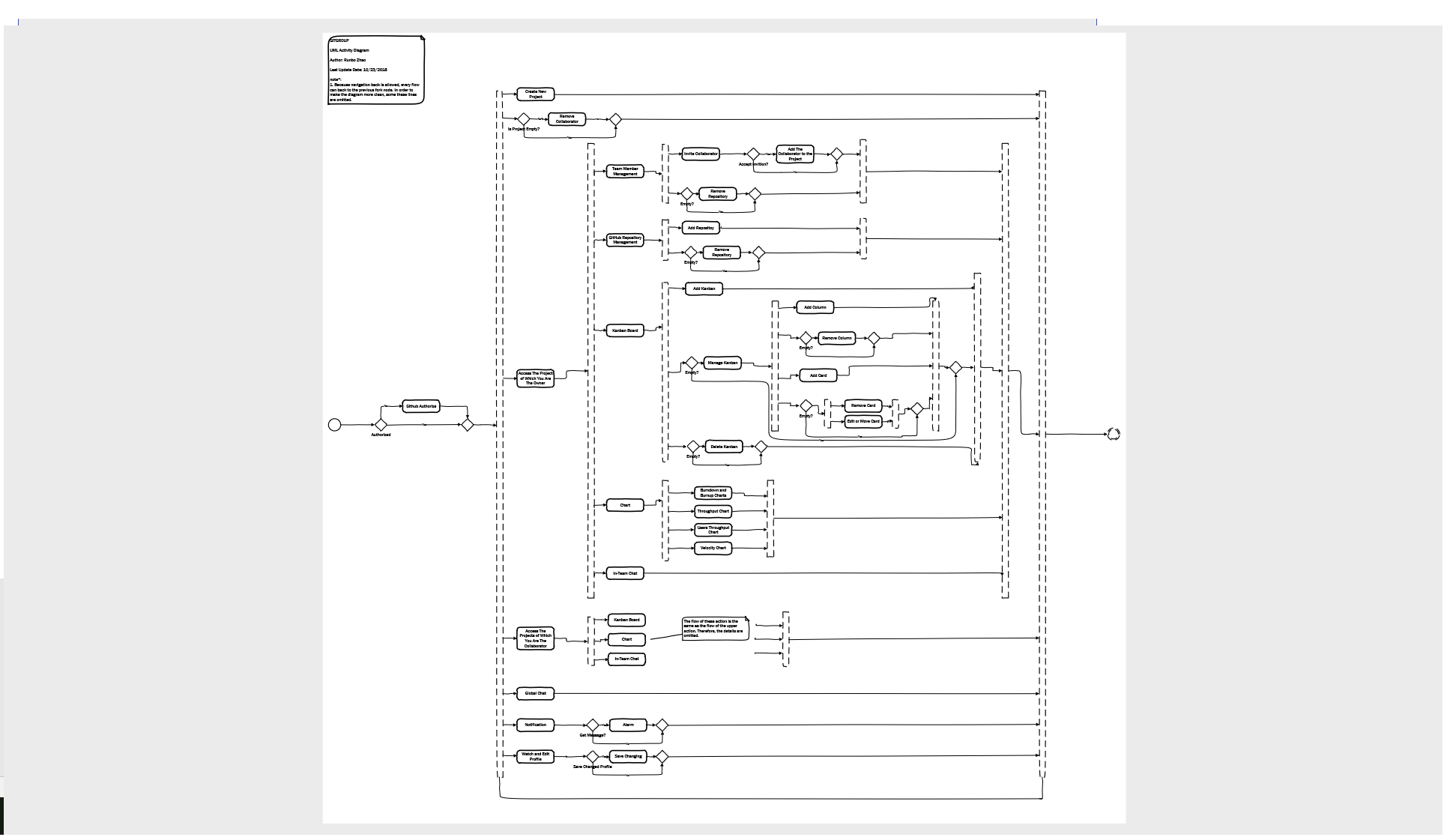
GitGroup API server get a temporary code GitHub server to back end server, and post information from back end to GitHub server in order to get access\_token.

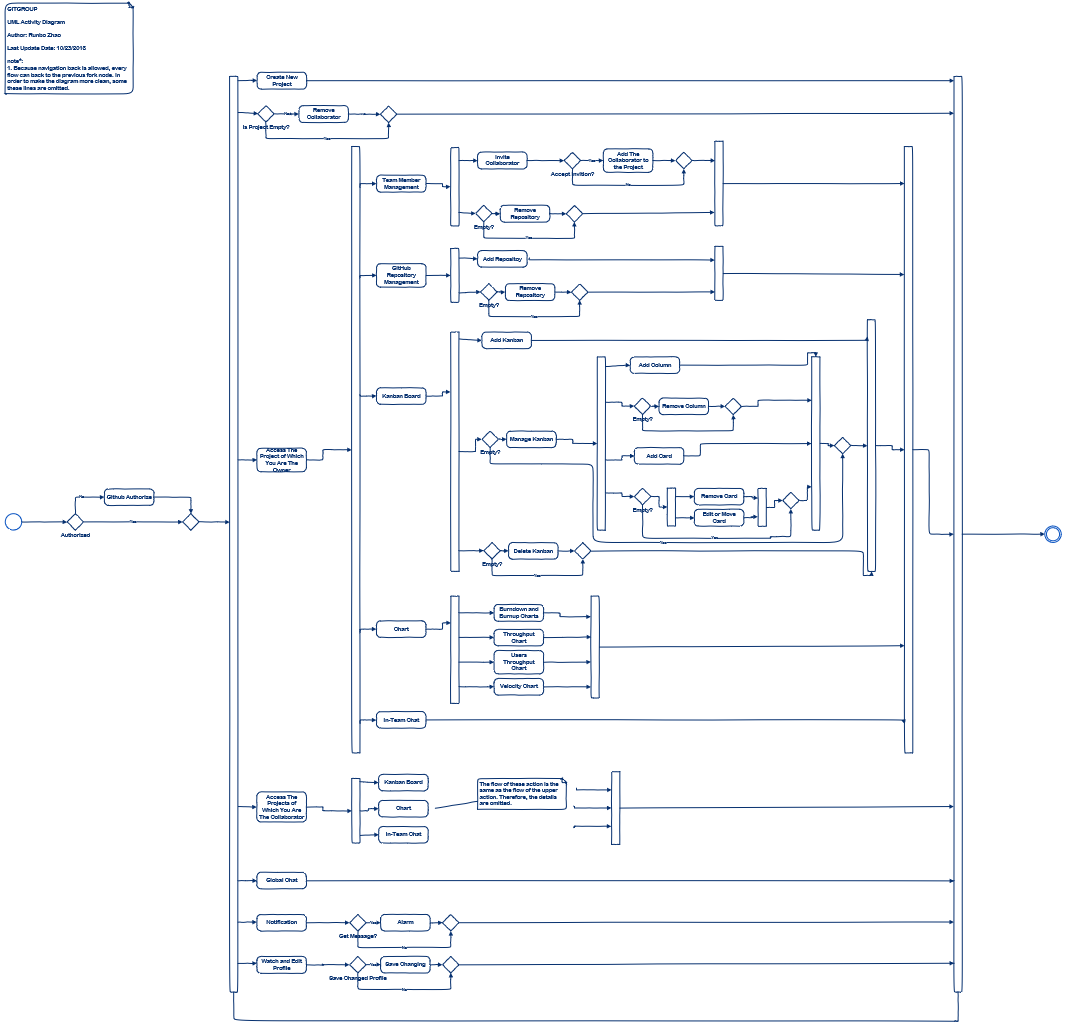
Then project Management Kanban Board Server access and manipulate data from GitHub Server and show on the Kanban board.

Chart data is got from GitHub server and accessed by project analysis – chart server.

Online chat

## Activity Diagram





## Class Diagram