Week2

Responsible for back-end technology selection, specification of coding requirements and selection of good dependencies. Define the back-end package structure and create the initial back-end project, then upload it to github, initialise the front-end project and upload it to github.

The master branch was the front-end project and the main project was the back-end project. The database design was discussed and the database design conventions were standardised (see Database.docx for details). Identified the modules that needed to be done and implemented the database table design and links, and created the mysql database.

Week 3

Completed jira backend task assignment, configured email to send captcha, implemented registration and login.

Created the JWT tool class to generate JWTs based on user IDs and parsed JWTs to return user IDs.

Completed the registration interface (detecting whether the verification code is correct and determining whether the mailbox has been registered) and generated token and userId for the front-end.

Completed the image verification code acquisition interface and the login interface. Finished writing the proposal together and submitted it successfully.

Week 4

创建Book Information， Homebook实体类，Mybatis。

完成了前后端的跨域问题，在后端实现，Spring MVC provides a cross-domain solution.

Cross-domain is when a browser requests a resource from one domain from a web page in another domain, and the domain name, port and protocol are any different. In a front and back-end separation model, the domain names of the front and back ends are not the same, and cross-domain access problems can occur at this point.

完成了本地缓存的配置，使用Caffeine. Caffeine 是 Java 8 对 Google Guava 缓存的重写，是一个提供了近乎最佳命中率的高性能的缓存库。

完成分布式缓存Redis的配置，本地缓存虽然有着访问速度快的优点，但无法进行大数据的存储。并且当我们集群部署多个服务节点，或者后期随着业务发展进行服务拆分后，没法共享缓存和保证缓存数据的一致性。 本地缓存的数据还会随应用程序的重启而丢失，这样对于需要持久化的数据满足不了需求，还会导致重启后数据库瞬时压力过大。

完成了可以查询homepage的books的接口，第一次从数据库中读入，但是后面可以从本地缓存中读出数据，加快了读取的速度。