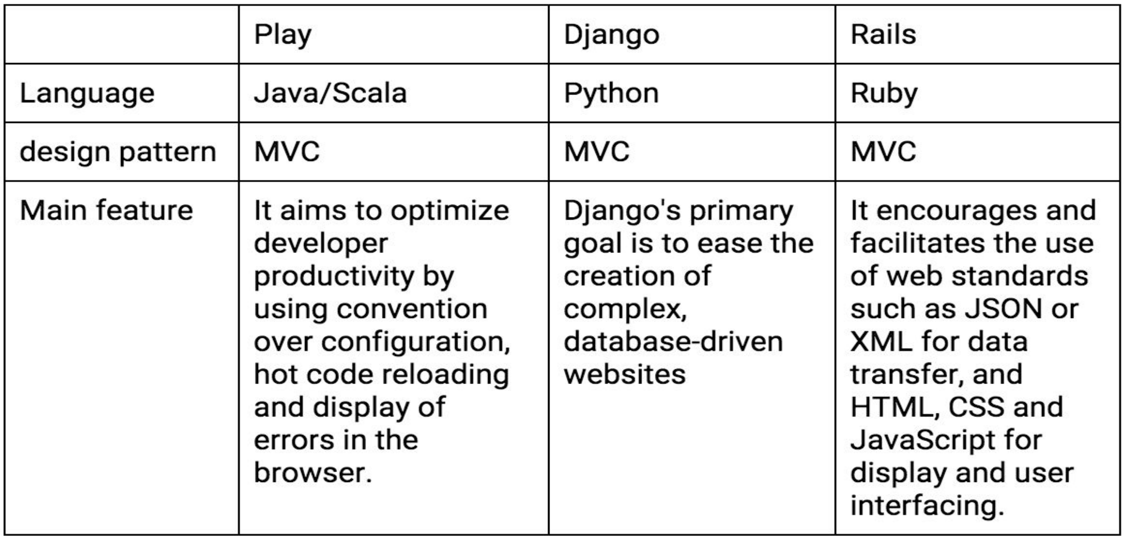
assumptions and considerations

We assume that this would be used by the whole company, so there would be lots of users and should support the multi-thread situation.

We have to consider all the choices we have and pick the best one of them
We have considered different web framework and database. We choose the Django and Postgres in the end.

Research on Web Development Framework

Play, Django and Ruby on rails are three of the most popular MVC (Model–view–controller) web development framework. These three popular web framework all has its own features , it is very hard to say which one is better than the others.



	Play	Django	Rails
Language	Java/Scala	Python	Ruby
design pattern	MVC	MVC	MVC
Main feature	It aims to optimize developer productivity by using convention over configuration, hot code reloading and display of errors in the browser.	Django's primary goal is to ease the creation of complex, database-driven websites	It encourages and facilitates the use of web standards such as JSON or XML for data transfer, and HTML, CSS and JavaScript for display and user interfacing.

We compare their features and our own project goal. Finally we decides to choose Django as our development framework because it is more friendly to database-driven websites and that is exactly what we need to store massive access control matrix. Django is a high-level MVC Python Web framework that make web development quick and tidy. It is also free and open source and helps developers to create complicated, database-driven websites.

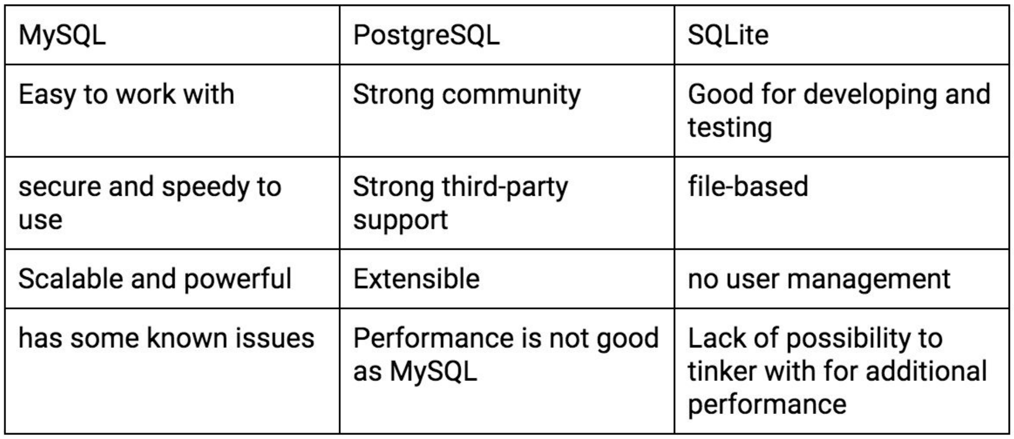
It can help us to finish building applications from scratch very fast. Django takes security seriously and can avoid many common security mistakes.

Its tutorial is all available online and easy to follow. That speeds up the developing process and makes sure deliverables of the project would be on time.

Research on Database

MySQL, PostgreSQL and SQLite are three of the most popular free open source database in web development.

They all have their advantages and disadvantages. So we compare them with our project goals in detail.



MySQL	PostgreSQL	SQLite
Easy to work with	Strong community	Good for developing and testing
secure and speedy to use	Strong third-party support	file-based
Scalable and powerful	Extensible	no user management
has some known issues	Performance is not good as MySQL	Lack of possibility to tinker with for additional performance

Because we want to make our application have reliability and data integrity, so we choose PostgreSQL as our final system database. In this way, we can ensure the integration of the system.

But during development process, we do not deal with massive amount of data. We tend to use SQLite as our development database, because it is lightweight small and easy to migrate.