# Approach Taken

For the team to collaborate effectively in this project we needed to agree on the language we would use to code the project, the tools we would use to produce the documentation and source code and the types of diagrams we want produced to complete the project. This section details how and why we came to the decisions we did and also looks at how we decided the design patterns and abstractions for the software.

## Technical

Tools that we would all have to use were decided by a vote at a formal meeting so that everyone got a say about what tools were used, however any tools that only one or two people would need to use were decided by those people and just reported on at the formal meetings. So some of the tools listed below are group decisions, some are individuals.

The project specification has required us to work in the Java programming language; this works out well as we have all had experience working with Java and are familiar with it, however we needed to choose and integrated development environment. Eclipse and vim were both suggested as environments for us to code the project; we decided on Eclipse as the majority of the team had more experience with it rather than vim. It was important that we used the same IDE as it would allow for easier integration of our different parts.

We then needed to choose what software tool we would use to produce the documentation. Microsoft Word 2010 was decided without much discussion as, again, it is something we are all familiar with and do not need to spend time learning and offers all the functionality we need. LaTeX was considered but as no one had enough knowledge using it we decided against that.

For the more informal documentation such as Agendas and the Time expenditure reports we decided to use Google Docs. As the team had worked together before we had experience using Google Docs together. It is useful for the time expenditure report as everyone can access the document and update their time as it happens rather than wait for the next meeting to let the Time Recorder know. It also allows for everyone to keep track on how many hours everyone else is putting in.

We decided on using Forge for our project repository. This is because we all had experience using it before and we no other better alternatives were suggested. It allows for us to add our work and download other team member’s work and creates a record of who did what in the project.

We will use [www.smartsheet.com](http://www.smartsheet.com) to produce a Gantt chart and keep it updated; Edward found this piece of software to be good at producing Gantt Charts. It will allow us to monitor if we are at the right stage of the project and what tasks need to be completed next. Keeping it updated will allow us to have a middle Gantt chart and a final Gantt chart to show how the work breakdown changed over the course of the project.

For the wireframe development that Oscar was focused on he was considering the use of Pencil Project initially but ended up using Balsamiq to produce initial wireframes to help the development of the View of the software.

?????UML diagrams will be needed in the requirements and analysis stages of the development. Gliffy was decided on by Edward and Brian that were looking at UML diagrams.

????J-Unit will be used for unit testing throughout the project to ensure that our software is relatively bug free. We chose J-Unit as it is free and everyone agreed.

The types of diagram we have chosen to include for our design of the software are, Use Case diagrams to identify all of the use cases from the requirements, class diagrams, so we have a way to break the software into smaller classes that can be coded and sequence diagrams to see how these classes are meant to interact within the system. There will also be wireframes so that we can design the UI of the system to be implemented in the later stages of development.

## Modelling

In the project specification one of the requirements it to structure the code using the Model-View-Controller architecture, so our design process will start with the architecture and move on down to the individual components and build up from that.