# Time Analysis

As a team, we initially allocated tasks to each member with an estimated time frame that it would take to complete. However, the actual time spent and activities they were spent on were quite different.

There are several reasons for this. Mainly, when we submitted Deliverables A and B, the time for a lot of tasks were incorrectly estimated, both underestimating and overestimating. Cases of overestimation mainly arose because we thought the task would be bigger than it turned out to be. For example, Ashwin was initially assigned 10 hours to write out the initial documentation and function stubs. In the end, he only did this task for approximately 1.5 hours. This was because a lot of what we would go on to do was still unknown at the beginning. As the project developed, team members wrote documentation and function stubs as they implemented required functionality.

For the case of underestimation, this arose from the team not realising the complexity of tasks when initially planning them. A case of this would be when Alex was assigned 4 hours to write code that was able to read .csv files. We found this task to be a lot harder than expected because it also depended on the .csv files having the correct format, which was the responsibility of the user. As such, checking for this correct input and handling exceptions added extra work that needed more time.

Another source of incorrect initial estimations and planning came from ‘specialisations’, that is, team members finding tasks they were good at and proceeding to carry it out. An example of this is Deliverables A. While initially assigned to the whole team, we found that Marcus’s previous experience from project management units allowed him to handle documentation with much greater efficiency than the rest of us. As a result, he proceeded to carry out tasks of this nature even if they were initially assigned to others. Apart from previous experience, some team members just took more interest in some sections. We realised that by allowing these team members to proceed with these tasks, they would be done to a higher standard instead of assigning them to someone who would not be interested.

In some cases, tasks that were initially planned became completely obsolete due to the varying scope of this project. The initial plan was to have several algorithms to process the client’s data. However, as we understood the data and the client’s request with more clarity, we realised that some of the algorithms were unnecessary. Cameron and Kieran were initially assigned the ‘player load’ algorithm, which was supposed to take 10 hours. Once we realised that this algorithm was actually unnecessary and furthermore, inefficient. This freed Cameron to help Alex on the Input/Output section and Kieran improved the graphical user interface. During this project, most team members found that they were more efficient when working with others. Due to this, a lot of tasks that were initially assigned to just one person were then split between two.

Specifically:

* Cameron and Alex worked together on Input/Output
* Dean and Ashwin worked on the GPS sections
* Marcus and Kieran worked on the graphical user interface

As these tasks required a significant amount of prior knowledge, it was efficient to assign fewer members per task as less people would have to read up on the background knowledge. Ensuring two people per task however meant that they could help each other out and this prevented members from getting stumped when it came to the harder sections.

On the whole, the biggest cause of the time differences in the planned and actual tasks resulted from the project changing scopes several times during the semester. Fortunately, all the team members were quite flexible and quick to adapt.

The following gives a summary of the tasks that were planned compared to the actual tasks per member:

## Kieran:

Chosen as the team’s representative when dealing with the client, Kieran also led the team admirably. Out of all the members, Kieran’s actual tasks matched up best with what he carried out. He spent most of his time designing the graphical user interface and helping other members out.

## Marcus:

While Marcus was initially supposed to design the graphical user interface, his experience was in QT which we found to be unsuitable for the project. However, his training from previous projects allowed him to efficiently deal with documentation. As can be seen from his timesheet, he spent most of his time creating and editing documents that were essential for keeping track of the team’s progress and as deliverables.

## Dean:

Early on, Dean research JUnit testing and related fields as Marcus and Kieran took care of the graphical user interface which Dean was assigned to. This research made him ideal for creating tests and collecting results throughout the project. Apart from this, Dean also spent a significant amount of time designing the classes to do with the GPS algorithms.

## Cameron:

Cameron was assigned the tasks of creating .vid files from the very start and as can be seen, he began this task as early as the fourth week. Additionally, he was assigned 10 hours to work on the player load heuristic and 10 more to work on optimising algorithms. He only spent about 3 hours on this as we found that the ‘player load’ heuristic was unnecessary. Instead, his time was better utilised working with Alex on input/output.

## Alex:

Alex and Cameron worked as a pair for most of the project. Alex was supposed to spend 20 hours on the GPS heuristic and the optimisation of algorithms. However, Dean and Ashwin were very interested in the GPS section and hence did most of the work on it. Alex instead focused on input/output, which took more time than planned.

## Ashwin:

While initially assigned 10 hours to work on documentation and function stubs, Ashwin only spent about an hour an a half on this task as we overestimated. The bulk of his time was spent working on the GPS heuristic, as he was very enthusiastic about it. Furthermore, even though he was assigned the tasks of conducting test suites, Dean took over this because he knew the most about conducting tests.

It was estimated that each team member would spend 60 hours on this project. At this point, the average time spent per member is approximately 38 hours. At our current rate, we do not think that we would have spent 60 hours by the end of this project and more likely the total time spent per member will be around 50 hours.