

# GP3 Presentation

Computer Science Development Exercise – Group 10  
School Of Computer Science – University Of Hertfordshire  
April 2020

# Team Working Roles & Responsibilities

- Roberto Figueiredo – Team Leader, Programmer and Design
- João Caeiro - Assistant Programmer and Documentation
- Manuel Rosmaninho - Database Design, Programmer and Tester
- Tiago Lima – UI Design and Presentation
- Mathew Chamberlain – Programmer and Documentation

# System Evaluation



## Performs Well

- Login System;
- Stage Entry System;
- Database data;
- Data creation;
- User centered workflow;
- Widgets;
- User dependent shortcuts;
- Design.



## Has bugs

- Logo disappears when login attempt fails;



## Not implemented

- Entry system Log(optional);
- Navigation guide;
- Date verification.

# Overall Analysis & Reflection

## What we did right:

- Balance distribution of tasks overtime;
- Teamwork version control software (GitHub);
- Implementation timeline;
- Planned implementation based on storyboards;
- All implementations made on branches requiring testing before merge to master.

## What we did wrong:

- Partial lack of communication;
- Not all members were knowledgeable on Git version control;
- Lack of technical knowledge to perform the tasks attributed resulting in unbalanced distribution of work;
- Better communication tools could have been used such as Slack.

# Overall Analysis & Reflection

Lessons learnt and how to overcome them in the future:

- **Unbalanced teams lead to unbalanced work**
  - When the team skills and knowledge is unbalanced it can lead to a great backlog and later on overload on other members. The best solution to the issue is ideally a balanced team with the same level of knowledge or a project with a lot of lower level tasks.
- **Organization and good communication is key for good implementation**
  - All team members must have in mind deadlines and targets not to create backlog, even more important when someone's task is dependent on it. It is also very important to let the team know what you're working on and assign issues to yourself when you take on them, so it doesn't lead to duplicate unnecessary work. A way to overcome this would be for mandatory reports during the week for 1 week sprint cycles.