Initial Report of Group CYCIN11PG

Project: **Team Pursuit Strategy Tool**

a1784310 Yiming Chen

a1789691 Lanxiao Li

a1832192 Penghao Ren

a1819238 Wing Yee Tong

a1702078 Kaifeng Xie

a1767091 Yuting Xu

a1795409 Yu Zhang

Project Vision

The current team pursuit strategy tools are mainly relying on the Worksheet runs on Microsoft Excel, and pure source code on MATLAB. This brings some challenges and inconvenience for the tools' major users, coaches. On the one hand, the worksheet and MATLAV source code do not have a friendly user interface, so it causes difficulties for maintenance and adjustment if user, the coach, lacks technical background. The other hand, current tools can only run locally, makes they are not portable and can not be accessed if user change the device. There is also some issue about analysis model. Current model's output is completely based on the input values that user given. If user wants to optimize an ideal strategy, it can only be done by adjusting plenty of variables by him/herself.

Our project is intended to design a web application for the strategy tool. Our first goal is to build the application based on the current strategy model and have a more user-friendly interface and can also access via Internet or local network. Which makes the front-end is easier to use by the coach. Meanwhile, the technicians can response the coach's requirements to maintain or customize the strategy model and update it from the back end. Our second goal is to use machine learning techniques improve the tools, to let it can help coach finding the optimized strategy by using the ML model.

Customer Q&A

During the kickoff meeting, the first thing we settled was that we wanted a web page over an application. Then we asked the client about whether we need to exclusively use MATLAB to incorporate optimization model into the web page. The client emphasized the necessary understanding of MATLAB and approved the use of other language in web application. Then together we came up with the initial design, that is to build a web application from scratch, translating/rewriting the former MATLAB codes into our preferred language, and implementing the optimization model based on the new codes. Further design like visualization and user-friendly functionality was not decided.

The first half of the kickoff meeting was mainly the client clarifying details and specific aspects of the project, and during the rest of the meeting, we finalized on some cornerstones of the project like language use and front-end framework. The reason why we did not cover and finalize a few more aspects of the project is that we were still weighing the different backgrounds of our members and cannot decide on more specific design of both front-end and back-end. For the next meeting, we should make sure that we gather necessary questions that will guide us for our upcoming steps before the meeting so that when we receive answers from the client, we can use them constructively in our following work.

A few follow-up questions that might appear in our next meeting are:

Should we keep the old functionality of a user-input database besides our new functionality? Do our users need it?

Should the optimization model include the optimization result for specific segments?

Should the result provide results more than visualization? E.g., suggestions.

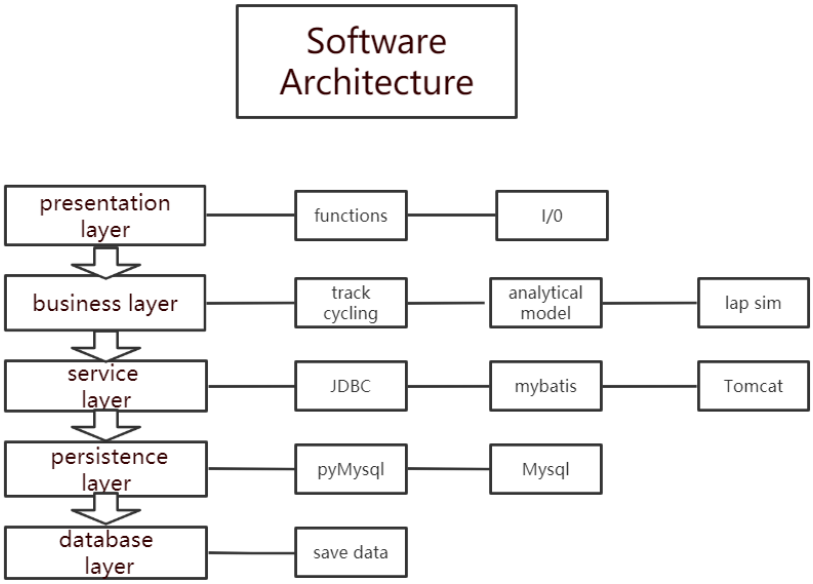
Users

Coaches are the main users of the project.

The game is a physical and psychological contest. Coaches not only need to pay attention to the physical ability of the athlete, but also the mental. on the physical side in the project, the coaches will understand the strengths and weaknesses of each athlete's body by the length of time of different combinations, thus helping coaches to designate different training programs for each athlete to play to their strengths and thus achieve the best interests of the group. And on the psychological side, due to the objectivity of the data, each athlete will understand the coach's organization and their own position in the team, thus reducing conflicts and increasing their confidence.

Moreover, coaches are constantly making decisions, whether it is in pre-game training, in-game strategy or post-game summaries. Therefore, we can't ask coaches to have enough time and energy to know all the related professional things such as sports physics and athletes' psychology. The project will provide coaches with a low-cost and low-time way to validate ideas or conclusions from academic professionals as advice to help them make their own decisions to achieve the results they need.

Software Architecture



Our team is trying to develop an application which can provide strategy to coaches by monitoring athlete's stats and other factors could affect the grade then processing. There are five layers in our architecture: presentation layer , business layer, service layer , persistence layer and database layer.

In presentation layer, we choose JSP and AJAX in order to providing real-time outputs to coaches during daily training. Analytical models have found widespread use in research and competitive cycling so building an accurate model is very important. All in all, presentation layer (front end) sends requests and instructions to business layer (back-end) then business layer processing the data and write into the database. Finally, the dynamic web page read the output and show it on dashboard.

Tech Stack and Standards

Our team made a discussion on the tech stacks we can use for the project of team pursuit strategy tool as shown below.

Front-end tech stack:

- HTML
- CSS
- JavaScript

Front-end frameworks:

- AngularJS
- Bootstrap
- jQuery

Back-end tech stack:

- Python

Back-end frameworks:

- Spring
- Django
- Laravel

For the front-end tech stack, the programming languages we have considered to apply for our project are HTML, CSS, and JavaScript. These are the fundamental programming languages to be used together to create and organize the contents of web pages and are suitable for our project. The front-end frameworks we have considered are AngularJS, Bootstrap, and jQuery. There are all free open-source frameworks based on the HTML, CSS or JavaScript.

Our team decide to choose python as our main back-end programming language, as several team members have experience in programming with python previously, and it is a popular

programming language with wide usage due to its convenience and efficiency, thus python will be a suitable programming language for our project. MATLAB will also be used for some times in our project, as we are required to deal with a given MATLAB code file. The frameworks we likely to use for back end include Spring, Django, and Laravel, which are all very popular back-end frameworks.

A Slack channel has been created for the team which allows team members to get contact with our client, and the team also created a Microsoft team which is mainly used for the discussions and communication within the team. These are the communication methods we will use throughout the project. The IDE to be used for the project is PyCharm, which is very easy for every team member to get access to.

Our team choose to follow the coding standards of PEP 8 during programming process, which is a style guide for python that provides information about code layout, comments writing, naming conventions, which help to enhance quality and readability of our codes.

Group Meeting and Team Member Roles

The team regular meeting will take place from 3pm to 4pm every Wednesday. Further meetings will be both held on face-to-face and online since the team has members offshore.

The next sprint retrospective meeting will be on 24th August, 2pm to 3pm before the regular meeting.

The team-members have set up a Microsoft Team channel for daily communication and document sharing. The team is also using the slack channel to hear feedback from the product owner.

The Scrum Master for upcoming 5 sprints through development cycle:

- Sprint 1: Yu Zhang (a1795409)
- Sprint 2: Wing Yee Tong (a1819238)
- Sprint 3: Penghao Ren (a1832192)
- Sprint 4: Kaifeng Xie (a1702078)
- Sprint 5: Yiming Chen (a1784310)

Snapshot

Project Backlog and Task Board

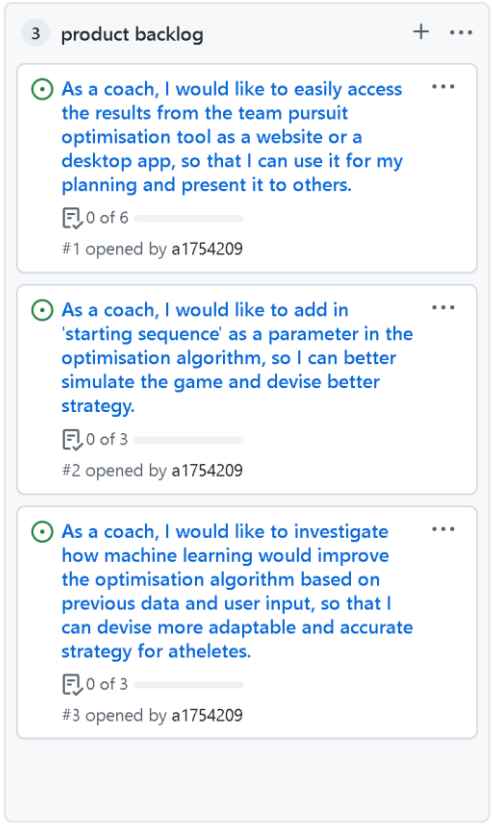


Figure 1. Project Backlog

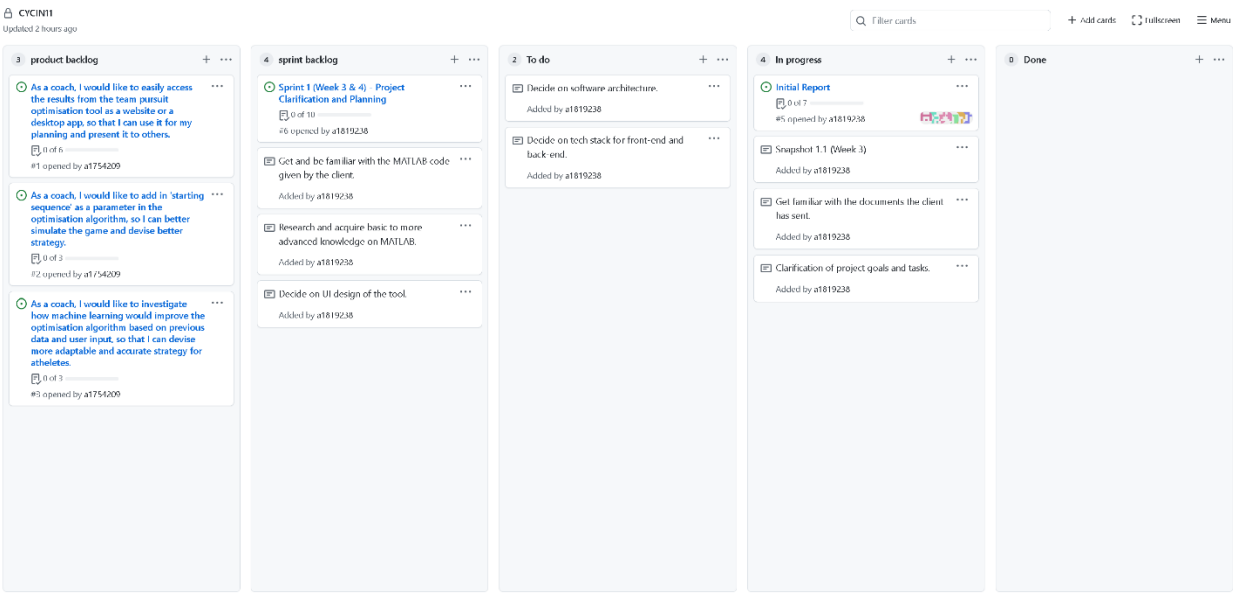


Figure 2. Task Board

Sprint Backlog and User Stories

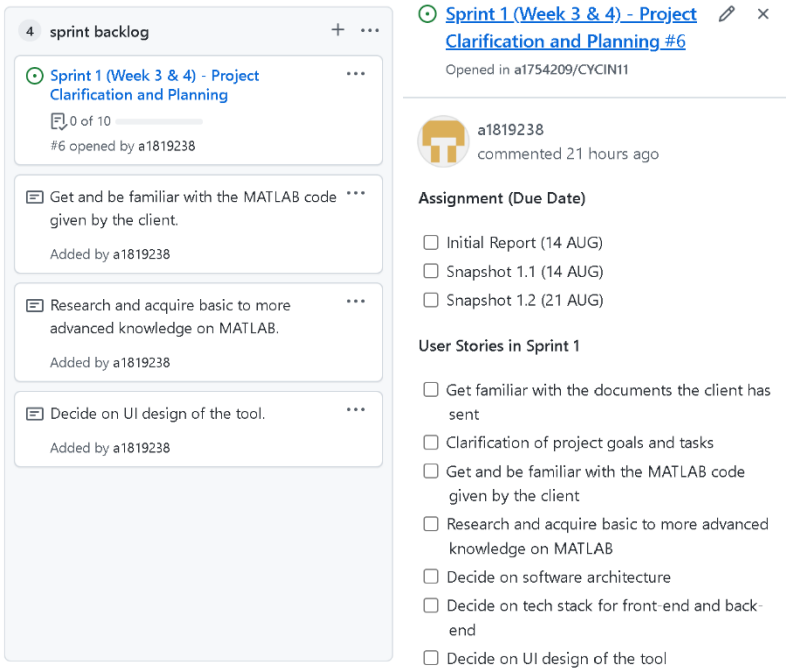


Figure 3. Sprint Backlog

- **Get familiar with the documents the client has sent**
    - After receiving the necessary documents from the client, team members are required to read these documents and understand basic project requirements. Additional team discussions or meetings will be arranged if any doubts or concerns arise.
    - ***Acceptance Criteria:*** All team members should have read the documents to understand the project requirements and specifications.
- **Clarification of project goals and tasks**
    - Gain mutual understanding of project goals and tasks on a team basis. Additional Q&A sections or meetings with supervisor will be arranged if necessary.
    - ***Acceptance Criteria:*** All team members are informed of and have necessary knowledge of the project goals and tasks.
- **Get and be familiar with the MATLAB code given by the client**
    - After receiving the MATLAB code from the client, team members should have gained access to the code and understand the basic idea of what the code does.
    - ***Acceptance Criteria:*** All team members should have read the codes given and conduct their own research to have a mutual understanding of basic ideas of the client's tool.
- **Research and acquire basic to more advanced knowledge on MATLAB**
    - Further research or advanced knowledge may be necessary for certain team members who are assigned MATLAB or machine learning related tasks. This user story is likely to continue through multiple sprints due to the code development nature of this project.
    - ***Acceptance Criteria:*** Responsible team members can acquire enough and necessary knowledge to work on the tasks assigned.
- **Decide on software architecture**
    - The team will research and decide on the best approach to complete the project in terms of software architecture. A flow chart or diagram to briefly describe said architecture is highly recommended.
    - ***Acceptance Criteria:*** The team has decided and finalized the tech stack for the project.

- **Decide on tech stack for front-end and back-end**
    - The team will research and decide on the best approach to complete the project in terms of tech stack (e.g., programming languages, IDE or other tools) front-end and back-end.
    - ***Acceptance Criteria:*** The team has decided and finalized the tech stack for the project.
- **Decide on UI design of the tool**
    - The team will design and decide on the best user interface design to fulfil the project requirements, including but not limited to aesthetics, user interactivity and tool functionality.
    - ***Acceptance Criteria:*** The team has designed a UI that is collectively reviewed and accepted by all team members.

Definition of Done

The following states the definition of done for the user stories applicable to Sprint 1:

- The sprint and related user stories were planned, notified, and accepted by all team members.
- Decision-type user stories are finalized and accepted by all team members.
- User stories listed in “To Do” or “In Progress” status that have met the acceptance criteria can be moved to the “Done” status.
- All user stories that have met the acceptance criteria are reviewed by at least 1 team member outside of the task-assigned team member or group.
- All user stories that have met the acceptance criteria are accepted by all team members.
- Any user story that hasn’t yet met the acceptance criteria in the current sprint is prolonged to the next or future sprint.

Summary of Changes

Not applicable. Being the first snapshot since the beginning of the project, we have added the user stories and related tasks necessary for the current sprint (i.e., Sprint 1 – Week 3 & 4).