

1. Home .....	2
1.1   Milestones .....	4
1.1.1   Sprint Plan .....	5
1.1.2 1   Sprint 1 Checklist .....	8
1.1.3 2   Sprint 2 Checklist (retrieved from Mauro) .....	10
1.1.4 3   Sprint 3 Checklist .....	14
1.1.5 4   Sprint 4 Checklist .....	18
1.1.6 Sprint Retrospectives .....	19
1.1.6.1 Sprint 1 Retrospective .....	20
1.1.6.2 Sprint 2 Retrospective .....	21
1.2   Specifications .....	22
1.2.1   Product Description .....	23
1.2.2   Motivational Model .....	25
1.2.3   User Stories .....	27
1.2.4   Persona .....	30
1.2.5   Prototype .....	37
1.2.6   Business Case for Cross-Team Collaboration .....	39
1.3   Development .....	41
1.3.1 Development Process .....	42
1.3.2 Version Control Guidelines .....	43
1.3.3 Test cases .....	44
1.3.4 Frontend workspace structure proposal .....	47
1.4   Resources .....	52
1.4.1   Contacts .....	53

# Home

## COMP90082 Hacking Materials User Interface (HA)

### Product overview

In computational materials engineering, engineering new materials are accelerated by avoiding expensive and lengthy experiments to demonstrate materials' performance. Data mining methods discover better materials by searching computer-generated databases with simulations predicting using high-throughput and high-performance computing. Unfortunately, the translation of data mining methods into engineering practice is still in its infancy with some frontrunners (e.g., Tesla). A barrier is materials engineers' software tools. An engineer with materials domain knowledge needs to access and process these data efficiently to make an informed decision for eventual machine learning strategies.

The [matminer](#) python library provides a framework to simplify the process of data retrieval, feature extraction, machine learning, and visualisation. But even a python tool is a high barrier in engineering practice. Our team product aims to provide a user interface for engineering for a standard process in mat miner: Retrieving data from databases, user selection of features to be extracted within the databases, performing simple machine-learning tasks (scikit learn, Keras), and visualising results.

### Goal

*Provide a user interface for standard processes in MatMiner: retrieving data from a database, users selecting features to be extracted from the database, performing simple machine learning tasks (Skikit Learn, Keras), and visualizing the results.*

### Team Members

Name	Dr Christian Brandl	Mauro Mello Jr			
Role	Client	Supervisor			
Name	<a href="#">Yanan Liu (Rep)</a>	<a href="#">Hongpei Lu</a>	<a href="#">Jiahao Ju</a>	<a href="#">Xinle Yu</a>	<a href="#">Rui ZHANG</a>
Role	Team Representative Product Developer	Product Developer	Product Developer	Product Developer	Product Developer
Student ID	1289747	1275238	1128182	1294310	1221568

### Recent space activity

[Jiahao Ju](#)

### Useful links

Trello	Github	Slack

### Tasks

- [Edit this home page](#) - Click *Edit* in the top right of this screen to customize your Space home page
- [Create your first page](#) - Click the *Create* button in the header to get started
- [Brand your Space](#) - Click *Configure Sidebar* in the left panel to update space details and logo
- [Set permissions](#) - Click *Space Tools* in the left sidebar to update permissions and give others access



10/08 | Meeting 12 08 , 2022 •

**Yanan Liu**

3 | Sprint 3 Checklist 08 , 2022 •



**Hongpei Lu**

09/23 | Backend Meeting 07 , 2022



**Mauro Mello Jr**

09/20 | Supervisor Meeting 8 05 , 2022 •



**Jiahao Ju**

Sprint 2 Retrospective 20 , 2022 •

# | Milestones

## Sprint

[| Sprint Plan](#)

[1 | Sprint 1 Checklist](#)

[2 | Sprint 2 Checklist \(retrieved from Mauro\)](#)

# Sprint Plan

Assigned to @Rui Zhang, [Xinle Yu](#), [Hongpei Lu](#)

Can download file in here: [Hacking\\_interface\\_Gantt Chart.pdf](#), see original work in [here](#).

**See details on Trello board [HERE](#).**

## Product Preparation (Aug 1 - Aug 7)

- Team form (Aug 1-3)
- Meeting with supervisor (Aug 4)
- Create a workspace on GitHub and Trello (Aug 5)
- Create Confluence pages (Aug 6)
- Draft of the workflow (Aug 7)

## Sprint 1: Inception (Aug 8 - Aug 21)

- Kick-off Meeting with Client (Aug 12)
- Motivational Model (Aug 8 - Aug 14)
- User Story & Personas Of Users (Aug 15 - Aug 21)
  - Student: Box Jelly
  - Industry: Blue Ring
  - Pro-user: Red Back
- Prototype (Aug 20 - Aug 21)
- Development Environment (Aug 15 - Aug 18)
- Development workflow (Aug 19 - Aug 21)
- Documentation (Aug 8 - Aug 21)

## Sprint 2: Development (Aug 22 - Sep 19)

- Retrospective of sprint 1 (Aug 22 - Aug 24)
- Sprint planning (Aug 23 - Aug 25)
- Sprint period (Aug 26 - Sep 19)
  - US 9 View and plot the results of the model
    - Frontend ([Rui ZHANG](#), 5 days)
  - US 22 See clear annotation or explanation of data points and features
    - Frontend ([Rui ZHANG](#), 5 days)
  - US 31 Select a Machine Learning model
    - Frontend ([Rui ZHANG](#), 2 days)
      - Machine learning ([Hongpei Lu Yanan Liu](#), 3 days each)
      - Backend ([Xinle Yu Jiahao Ju](#), 5 days each)
  - US 33 Browse built-in ML models
    - Frontend ([Rui ZHANG](#), 2 day)
      - Machine learning ([Hongpei Lu Yanan Liu](#), 10 days each)
      - Backend ([Xinle Yu Jiahao Ju](#), 5 days each)
  - Testing
    - Frontend ([Rui ZHANG](#), 3 days)
    - Machine learning ([Hongpei Lu Yanan Liu](#), 3 days each)
- Task Tracking (Aug 22 - Sep 19)
- Documentation (Aug 22 - Sep 19)

## Sprint 3: Development (Sep 20 - Oct 21, TBC)

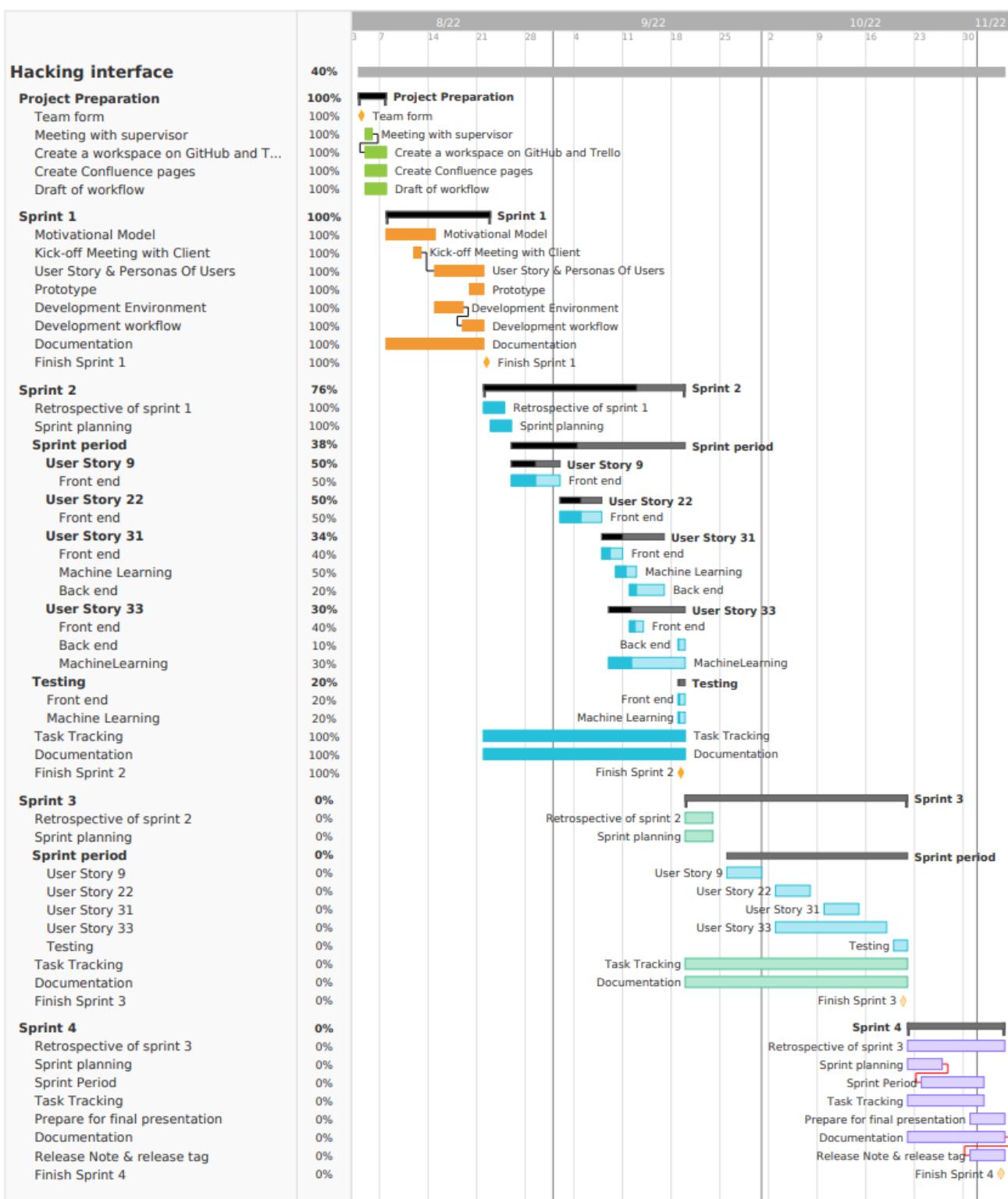
- Retrospective of sprint 2 (Sep 20 - Sep 23)
- Sprint planning (Sep 20 - Sep 23)
- Sprint period (Sep 24 - Oct 21)
  - US 8 Use different types of plotting graphs
    - Frontend ([Rui ZHANG](#), estimated 10 days)
  - US 9 View and plot the results of the model
    - Frontend ([Rui ZHANG](#), estimated 3 days)
      - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 3 days each)
      - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
  - US 31 Select a Machine Learning model
    - Frontend ([Rui ZHANG](#), estimated 5 days)
      - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 3 days each)
      - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
  - US 33 Browse built-in ML models
    - Frontend ([Rui ZHANG](#), estimated 5 days)
      - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 5 days each)

- Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
- US 39 Select split ratio of data
  - Frontend ([Rui ZHANG](#), estimated 5 days)
  - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 5 days each)
  - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
- Testing
  - Frontend ([Rui ZHANG](#), estimated 5 days)
  - Machine learning ([Hongpei Lu Yanan Liu](#), estimated 5 days each)
  - Backend ([Xinle Yu Jiahao Ju](#), estimated 5 days each)
- Task Tracking (Sep 20 - Oct 21)
- Documentation (Sep 20 - Oct 21)

## Sprint 4: Product (Oct 22 - Nov 4, TBC)

- Retrospective of sprint 3 (Oct 22 - Oct 26)
- Sprint planning (Oct 22 - Oct 26)
- Sprint period (Oct 24 - Nov 1)
  - Testing
- Task Tracking (Oct 22 - Nov 4)
- Prepare for final presentation (Oct 31 - Nov 4)
- Documentation (Oct 22 - Nov 4)
  - Release Note & release tag (Oct 31 - Nov 4)

## Gantt Chart



# 1 | Sprint 1 Checklist

This checklist helps double check work for Sprint 1.

## Background description, client goals, motivation

- Product overview, background and goals were created.
- DO-BE-FEEL list and GOAL MODEL were created.
- The goal model is consistent with the client understanding of the problem and with DO-BE-FEEL list.

## Personas

Make sure that your Personas satisfies the following criteria:

- 2-3 personas were developed to help with requirements validation.
- Personas are based on the research done by students and the discussion with industry partners.
- Personas are inclusive and diverse.

## Analysis of requirements (User Stories or Use Cases)

- The analysis of requirements was performed on most of the existing requirements.
- The [new set of] requirements is consistent to the scope of the product, completely cover the new capabilities required by the client and are well documented/structured/organized on Confluence.
- The requirements can be documented in the form of user stories or use cases, supplementary specification of design/implementation/deployment requirements, prototypes, and others. It may also be necessary to be explicit about what is not in scope to define the scope boundary more clearly.

## Development environment

- Confluence is organized (cover page, product details, requirements, technical details about the product, meeting minutes and so on).
- Trello (or GitHub product or JIRA) is created, structured and organized.
- README file is updated and provide details about the product, workflow (branches/naming conventions and so on).

## Plan

- A plan (or discussion on what to do next) was provided (requirements to develop, technologies to use, infrastructure to deploy the product) for Sprint 2 and Sprint 3.
- Requirements were estimated and prioritized.
- Backlog items can be found in Trello (or GitHub product or JIRA).

## Meetings

- Meetings are recorded in Confluence and only. They were NOT exported to GitHub as they're the part of internal process.

## GitHub

- Folders are structured (On Canvas, visit Assignment -> "Sprint 1: Confluence Space, product background and elicitation documents" page: you can find requirements for folders' structure.)
- Sprint 1 documents were exported from Confluence and added to the repository (and are updated)
- README file is updated and explains the team's repository
- A baseline tag was generated for this Sprint (On Canvas, visit Assignment -> "Sprint 1: Confluence Space, project background and elicitation documents" page: you can find requirements for the baseline tag)

## Additional Information

*do you have any other additional information you'd like to share with us? Please add it here.*

## 2 | Sprint 2 Checklist (retrieved from Mauro)

### Versions

Version ID	Description	Editor	Date
1.0	Combined checklist from inputs of the supervisor and all three teams	<a href="#">Yanan Liu</a>	2022-9-9
1.1	Had a walk through and checked what's done	@all	2022-8-15
1.2	Add task allocation and highlighted some unclear items	<a href="#">Rui ZHANG</a>	2022-9-16

### Version 1.2

Status	Tasks
<input checked="" type="checkbox"/>	1. Presentation – spelling and grammar, names ---- <a href="#">Rui ZHANG Hongpei Lu</a>
<input checked="" type="checkbox"/>	Spelling: full document checked
<input checked="" type="checkbox"/>	Grammar: full document checked
<input checked="" type="checkbox"/>	Company and product names, trademarks, professional titles, technical terms, acronyms, and abbreviations checked (in full first time, abbreviated thereafter, capitalization, spelling, trademarks, etc.): full document checked
<input checked="" type="checkbox"/>	Unit names and use: full document checked (e.g., AUD500 or A\$500, 16 MB, 16 Mbit/sec, etc.)
<input checked="" type="checkbox"/>	Contact UoM IT services to find out the correct names for cloud services/servers/platform
<input checked="" type="checkbox"/>	2. Formatting and visual presentation (mostly Confluence) ---- <a href="#">Rui ZHANG Hongpei Lu</a>
<input checked="" type="checkbox"/>	2.1. Content presentation
<input checked="" type="checkbox"/>	Content produced by merging artifacts from different teams is presented in a consistent way across all documents
<input checked="" type="checkbox"/>	Smaller icons (or links) for Slack, Trello, Github and other products
<input checked="" type="checkbox"/>	Content of pages fits onto Confluence export page (resize if required)
<input checked="" type="checkbox"/>	* Add document description including file name/purpose/changelog
<input checked="" type="checkbox"/>	Make sure titles are detailed and avoid using general-purpose titles.

<input checked="" type="checkbox"/>	It is clear which teams were responsible for which deliverables
<input checked="" type="checkbox"/>	2.2. Headers and footers
<input checked="" type="checkbox"/>	Confluence pages header and footer reflect current sprint
<input checked="" type="checkbox"/>	Confluence exported pages include header and footer
<input checked="" type="checkbox"/>	3. Preparing and packaging documents for submission <a href="#">Yanan Liu</a> <a href="#">Xinle Yu</a> <a href="#">Rui ZHANG</a>
<input checked="" type="checkbox"/>	Tag and resources for current sprint verified and validated
<input checked="" type="checkbox"/>	Assessments page should not be included in the exported documentation
<input checked="" type="checkbox"/>	* Filter and rename Assessments Page so that it does not include references to course-related terminology
<input checked="" type="checkbox"/>	Prior versions of artifacts should not be included in the Confluence export for the current sprint -waiting for final version
<input type="checkbox"/>	4. Sprint-specific items (sprint 2)
<input checked="" type="checkbox"/>	4.1. Confluence (infrastructure) ---- <a href="#">Hongpei Lu</a>
<input checked="" type="checkbox"/>	Clear, easy to navigate structure with indicative page names (see items 1–3 above)
<input checked="" type="checkbox"/>	Minimize depth (more on each page, fewer clicks)
<input checked="" type="checkbox"/>	No multiple versions of content (handled by built-in versioning)
<input checked="" type="checkbox"/>	Visible and editable content (e.g., minutes saved directly, diagrams can be updated without referring to an external editor)
<input checked="" type="checkbox"/>	4.2. Confluence (project contents and consistency) ---- <a href="#">Rui ZHANG</a> <a href="#">Xinle Yu</a> <a href="#">Jiahao Ju</a>
<input checked="" type="checkbox"/>	* Project []description, []scope, []stakeholders, []team members, []team roles with [ ]responsibilities, []technology used, [ ]user stories, []test cases
<input checked="" type="checkbox"/>	Weekly team meeting with agenda, meeting minutes, decisions, action items

<input checked="" type="checkbox"/>	Sprint planning, sprint review, and sprint retrospective meetings for each sprint
<input checked="" type="checkbox"/>	* Stand-up meetings at least twice per week (documented on Confluence only)
<input checked="" type="checkbox"/>	* Product and project-related communication from the client (documented on Confluence only)
<input type="checkbox"/>	4.3. Task tracking ---- <a href="#">Jiahao Ju</a> <a href="#">Xinle Yu</a> <a href="#">Yanan Liu</a>
<input checked="" type="checkbox"/>	Product backlog along with a sprint backlog
<input checked="" type="checkbox"/>	* Tasks in sprint backlog [ ]with adequate level of granularity and [ ]linked to their user stories
<input checked="" type="checkbox"/>	Tasks in sprint backlog include effort estimate and due date
<input checked="" type="checkbox"/>	Tasks in sprint backlog include dependencies to other tasks
<input checked="" type="checkbox"/>	Team members assigned to tasks and responsible for moving them along the lanes of the board in a timely manner
<input checked="" type="checkbox"/>	Documented estimation process (in Confluence)
<input checked="" type="checkbox"/>	Documented workflow (in Confluence)
<input type="checkbox"/>	4.4. Version control @all
<input checked="" type="checkbox"/>	Good quality code: coding standard described (in Confluence) and followed
<input checked="" type="checkbox"/>	Test code and cases clearly identified
<input checked="" type="checkbox"/>	Instructions or infrastructure to build and run the product, including running test cases
<input checked="" type="checkbox"/>	Sensible log messages in all commits, including references to tracked issues
<input checked="" type="checkbox"/>	Documented commit and branching policy (in Confluence)
<input checked="" type="checkbox"/>	Well-defined code review policy (reviewers, timeframes, checklist for review such as this one)

<input checked="" type="checkbox"/>	Sensible policy on tagging for system versions (milestone versions, ability to repeat tests)
<input type="checkbox"/>	4.5. Product @all
<input checked="" type="checkbox"/>	Product outcomes for sprint 2 commensurate with project difficulty
<input checked="" type="checkbox"/>	Outcomes are of a standard, structure and content clearly visible to client and assessors
<input type="checkbox"/>	* Product is deployed --not ready until sprint 4 --need conform with other teams
<input type="checkbox"/>	4.6. * GitHub ---- Xinle Yu Yanan Liu
<input checked="" type="checkbox"/>	* Folders are structured
<input checked="" type="checkbox"/>	* Sprint 2 documents [ ]updated, [ ]exported from Confluence, [ ]added to repository --need conform with other teams
<input checked="" type="checkbox"/>	* README file updated [ ]explains team's repository and [ ]new release --need conform with other teams
<input checked="" type="checkbox"/>	* Baseline tag [ ]generated, [ ]resources confirmed --need conform with other teams

# 3 | Sprint 3 Checklist

This checklist helps you double check your work for Sprint 3.

## 1. Verification and Validation (V&V)

### 1.1. Sprint 3 checklist V&V

1.1.1. All BlueRing team members reviewed and know their...

- [ ] roles
- [ ] assigned activities
- [ ] buddy or stand-in

- - - - -

1.1.2. All BoxJelly team members reviewed and know their...

- [ ] roles
- [ ] assigned activities
- [ ] buddy or stand-in

- [ ] Each assignee adds [due date and time] to check items owned

- - - - -

1.1.3. All RedBack team members reviewed and know their...

- [ ] roles
- [ ] assigned activities
- [ ] buddy or stand-in

- [ ] Each assignee adds {due date and time} to check items owned

- - - - -

1.2. Mon 10/10/2022 22:00: Dry run prior to sprint 3 deliverables submission and final date for making \*any\* changes to the checklist (sent to MMJ)

### 1.2.1. BlueRing

- [ ] Completed dry run of all submission activities prior to pressing Upload
- [X] Changes to checklist, if any, attached and returned to MMJ

### 1.2.2. BoxJelly

- [ ] Completed dry run of all submission activities prior to pressing Upload
- [ ] Changes to checklist, if any, attached and returned to MMJ

### 1.2.3. Redback

- [ ] Completed dry run of all submission activities prior to pressing Upload
- [ ] Changes to checklist, if any, attached and returned to MMJ

- - - - -

## 2. General items

### 2.1. Presentation – spelling and grammar, names

- [ ] Spelling: full document checked
- [ ] Grammar: full document checked

- [ ] Company and product names, trademarks, professional titles, technical terms, acronyms and abbreviations checked (in full first time, abbreviated thereafter, capitalization, spelling, trademarks, etc.): full document checked

- [ ] Unit names and use: full document checked (e.g., AUD500 or A\$500, 16 MB, 16 Mbit/sec, etc.)

- [ ] Contact UoM IT services to find out the correct names for cloud services/servers/platform

- - - - -

### 2.2. Formatting and visual presentation (mostly Confluence)

- [ ] Content produced by merging artifacts from different teams is presented in a consistent way across all documents
  - [ ] Smaller icons (or links) for Slack, Trello, Github and other products
  - [ ] Content of pages fits onto Confluence export page (resize if required)
  - [ ] Add document description including file name/purpose/changelog
  - [ ] Make sure titles are detailed and avoid using general-purpose titles.
  - [ ] Confluence pages header and footer reflect current sprint
  - [ ] Confluence exported pages include header and footer
- 

### 2.3. Preparing and packaging documents for submission

- [ ] Tag and resources (exported Confluence package) for current sprint verified and validated by the QA lead and one other team member
  - [ ] Assessments page should not be included in the exported documentation
  - [ ] Filter and rename Assessments Page so that it does not include references to course-related terminology
  - [ ] Prior versions of artifacts should not be included in the Confluence export for the current sprint
- 

### 2.4. It is clear which teams were responsible for which deliverables

- [ ] There is a deliverables summary list separate from the process action items in the checklist
- 

### 2.5 Delivery mechanism for each item is known and assigned to team members with due dates

- [ ] Confluence: team members allocated
  - [ ] Confluence: due date confirmed
  - [ ] GitHub: team members allocated
  - [ ] GitHub: due date confirmed
- 

## 3. Sprint-specific items (sprint 3)

### 3.1. Confluence (infrastructure)

- [ ] Clear, easy to navigate structure with indicative page names (see item 2 above)
  - [ ] Minimize depth (more on each page, fewer clicks)
  - [ ] No multiple versions of content (handled by built-in versioning)
  - [ ] Visible and editable content (e.g., minutes saved directly, diagrams can be updated without referring to an external editor)
- 

### 3.2. Confluence (project contents and consistency)

#### 3.2.1. Project definition

- [ ] description
  - [ ] scope
  - [ ] stakeholders
  - [ ] team members
  - [ ] team roles with
  - [ ] responsibilities
  - [ ] technology used
  - [ ] user stories
  - [ ] test cases
-

### 3.2.2. General activities

- [ ] Weekly team meeting with agenda, meeting minutes, decisions, action items
  - [ ] Sprint planning, sprint review and sprint retrospective meetings for each sprint
  - [ ] Stand-up meetings at least twice per week (documented on Confluence only)
  - [ ] Product and project-related communication from client (documented on Confluence only)
  - [ ] Recordings and notes from knowledge sharing meetings are documented in Confluence
  - [ ] Team representatives meeting notes include review of open PRs (to enforce 24/48h code review deadline) (Use filter: - `is:pr -label:DRAFT is:open`)
- - -

### 3.3. Task tracking

- [ ] Product backlog kept along with a (low-level) sprint backlog
- [ ] Tasks in sprint backlog have adequate (low) level of granularity
- [ ] Tasks in sprint backlog are linked to their user stories
- [ ] Tasks in sprint backlog include effort estimate and due date
- [ ] Tasks in sprint backlog include dependencies to other tasks

Team members assigned to tasks and responsible for moving them along the lanes of the board in a timely manner

- [ ] Team members assigned to tasks
  - [ ] Due dates assigned to tasks
  - [ ] Documented estimation process (in Confluence)
  - [ ] Documented workflow (in Confluence)
- - -

### 3.4. Version control

- [ ] Good quality code: coding standard described (in Confluence) and followed throughout
- [ ] Test code and cases clearly identified
- - - - -
- [ ] Instructions or infrastructure to build and run the product
- [ ] Instructions or infrastructure to run test cases
- [ ] Sensible log messages in all commits, including references to tracked issues
- [ ] Documented commit and branching policy (in Confluence)
- - - - -

### 3.5. Well-defined code review policy

- [ ] reviewers
  - [ ] timeframes
  - [ ] checklist for review (such as this one)
- - - - -

### 3.6. Sensible policy on tagging for system versions

- [ ] milestone versions
  - [ ] ability to repeat tests
- - - - -

### 3.7. Product

- [ ] Product outcomes for sprint 3 commensurate with project difficulty
- - - - -

#### 3.7.1. Outcomes of product show...

- [ ] Clear and demonstrable use of standards throughout project

- [ ] Structure and content clearly visible (relevant) to client
  - [ ] Structure and content clearly visible (relevant) to assessors
- - - - -

3.7.2. Each requested/proposed functional and non-functional feature can be...

- [ ] Demonstrated to client
  - [ ] Demonstrated to assessors
  - [ ] Assessed for feedback
- - - - -

- [ ] Product is deployed

- - - - -

### 3.8. GitHub

- [ ] GitHub folders are structured
- - - - -

3.8.1. GitHub sprint 3 documents

- [ ] Updated
  - [ ] Exported from Confluence
  - [ ] Added to repository
- - - - -

3.8.2. GitHub README file updated to...

- [ ] Explain team's repository
  - [ ] New release
- - - - -

3.8.3 GitHub baseline tag...

- [ ] Generated
- [ ] Resources confirmed
- [ ] Courtesy link sent to MMJ after submission (which is well before the deadline)

# 4 | Sprint 4 Checklist

This checklist helps you double check your work for Sprint 4.

## Meetings

Make sure your meetings (team meetings, supervision meetings and meetings with industry partners) are documented in Confluence (and only).

- Meetings are recorded in Confluence and only. They were NOT exported to Github as they're part of internal process.

## Release notes

Make sure that:

- Students created a clear and complete document explaining the client how to access developed resources, how they are organised on Github, and also provided details on how to run their project (digital prototypes).

## Release TAG

Make sure that:

- Students generated a release TAG on Github (containing all project resources, including exported documents/diagrams from Confluence).

## ZIP File

Make sure that:

- Students created an organised release to the client on Github, including: documents, tests, data samples, prototypes, and images.
- Release was downloaded from Github, packed in a ZIP file, sent to the client (together with release notes) and added to Confluence (under Handover page).

## Final Presentation Slides

Make sure that:

- Added to Confluence and Github. Industry partner will receive it as part of final release package.

## High Fidelity Digital Prototype and Data Sample

Make sure that:

- Industry partner was granted access to the final version of digital prototype and can run it independently after the end of the teaching semester.
- A document was also generated with SAMPLE DATA industry partner will need to simulate tasks and scenarios in the final digital prototype.

## Self-reflection

Self-reflection is one document for the whole team (with one subsection for each student and a final one for the whole team explaining how they addressed received feedback in the project).

Make sure that the self-reflection contains:

- 350 words maximum per student
- Self-reflection about students' journey in the subject and brief discussion on what & how they addressed received feedback after usability tests with clients.

## Additional Information

*do you have any other additional information you'd like to share with us? Please add it here.*

# Sprint Retrospectives

[Sprint 1 Retrospective](#)

# Sprint 1 Retrospective

What went well	What could be done better	What we think of the client interactions
<ul style="list-style-type: none"> <li>An important decision was made to merge the three project groups.</li> <li>Within the group, the division of labor is clear, the atmosphere is harmonious, and the assigned task is completed in time.</li> <li>The three groups learned and discussed with each other, which was of great significance for the progress of the project and the completion of Confluence.</li> <li>Confluence page is neat and beautiful.</li> <li>The communication with the Client was very harmonious, and the Client was highly satisfied with the progress of the three teams.</li> <li>The Supervisor plays a key role in guiding and supervising the progress of the project and the development of thinking.</li> <li>The use of Trello has played a positive role in the division of tasks and Sprint Plan.</li> <li>Shared GitHub repository and Trello board.</li> <li>Visibility of other team's Confluence pages.</li> </ul>	<ul style="list-style-type: none"> <li>The discussion on Slack is not active enough.</li> <li>The process of deciding on three groups and developing the project took too much time.</li> <li>In the meeting of three groups, the group members were not active enough to speak.</li> <li>There were mistakes with the details on some Confluence pages.</li> <li>Our team and BoxJelly team confused about what each team should do next and what other teams are doing.</li> <li>No regular standup meeting.</li> <li>Didn't prepare a way to track everybody's work.</li> </ul>	<ul style="list-style-type: none"> <li>We believe that Dr. Christian Brandl, as a Client of the project, is very cooperative in the progress of the project.</li> <li>He actively answered our questions and put forward his requests and ideas in the group meeting.</li> </ul>

# Sprint 2 Retrospective

What went well	What could be done better	What is confusing

# | Specifications

- | Product Description
- | Sprint Plan
- | Motivational Model
- | User Stories
- | Persona
- | Prototype
- | Business Case for Cross-Team Collaboration

# | Product Description

## Background

- In computational materials engineering, the engineering design of new materials is accelerated by avoiding costly and lengthy experiments to demonstrate the properties of materials. Data mining methods discover better materials by searching computer generated databases and making simulation predictions using high throughput and high-performance computing.
- Unfortunately, for some of the first movers, such as Tesla, translating this new thinking into engineering practice is still in its infancy. One obstacle is material engineers' software tools.
- Engineers with knowledge of the material domain need to effectively access and process this data in order to make informed decisions for the final machine learning strategy.

## Goal

- Provide a user interface for standard processes in MatMiner: retrieving data from a database, users selecting features to be extracted from the database, performing simple machine learning tasks (Scikit Learn, Keras), and visualizing the results.

## Sponsor

- **Dr Christian Brandl**
- Lecturer and sponsor at the University Of Melbourne.
- Completed his Ph.D. in Materials Science & Engineering at the École Polytechnique Fédérale de Lausanne EPFL and the Paul Scherrer Institute (Switzerland), followed by a post-doc in the Theoretical Division at the Los Alamos National Laboratory (USA).
- His research focuses on predictive atomistic simulation approaches to enable rational materials design.

## In-scope

The product is to design and build the following:

- A web page tool to help professional user and regular user or student who has no or very little experience in machine learning analysis method to complete material analysis;

Users requirements:

- Detailed analysis of user requirements;
- Design a beautified, user-friendly interface;
- Design a login page;
- Provide hints and guidance for new users;
- Design various machine learning models in Python that allow users to select features from a database and obtain a visual result;
- Allow to add new features;
- Provide code interface for pro user;
- The workflow allows to download;
- Machine learning model accuracy reaches a sufficient level;
- Complete the improvement suggestions that may be put forward by the clients;
- At least allow 30+ users to operate concurrently;
- Report the progress and results of the product;
- Runs on Melbourne Research Cloud;
- The final product results passed the test.

## Out-of-Scope

- The team has no obligation to release back-end code to the public;
- Operating maintenance is not within the scope of the team's obligations;
- The interpretability of the machine learning model does not need to be guaranteed, which means that the team members can choose the black box model as the machine learning model;
- Due to the lack of resources, the backend server cannot guarantee the stability of multiple simultaneous access.

## Stakeholders

Name	Position	Internal/External	Product Role
Dr. Christian Brandl	The person who originally wanted to start this product	External	<b>Product Client</b>
Mauro Mello Jr	The person who directs and supervises product team members on the product	Internal	<b>Product Supervisor</b>

Yanan Liu	Student of the University Of Melbourne enrolled in COMP90082	Internal	<b>Team Representative</b> <b>Product Developer</b>
Hongpei Lu	Student of the University Of Melbourne enrolled in COMP90082	Internal	<b>Product Developer</b>
Jiahao Ju	Student of the University Of Melbourne enrolled in COMP90082	Internal	<b>Product Developer</b>
Xinle Yu	Student of the University Of Melbourne enrolled in COMP90082	Internal	<b>Product Developer</b>
Rui Zhang	Student of the University Of Melbourne enrolled in COMP90082	Internal	<b>Product Developer</b>

## Techniques

- Web Development Techniques
  - NodeJS is a back-end JavaScript runtime environment for building scalable network applications. In this product, it is applied to build the structure of the online tool.
- Frontend
  - React is a JavaScript library for building user interfaces. It is applied to achieve designed functions.
- Backend
  - Django is a Python-based web framework. The backend process and management are based on it.
- Machine Learning
  - [MatMiner](#) is a Python library for data mining the properties of materials, the feature extraction from database will be achieved based on it.
  - [Scikit Learn](#) is a Python library for Machine Learning with various algorithms. It is used for predicting properties of materials.

# Motivational Model

Assigned to [Yanan Liu](#), [Hongpei Lu](#)

## Versions

Version ID	Description	Editor	Date
1.0	First version based on initial understanding of the product and first client meeting	<a href="#">Yanan Liu</a>	2022-8-17
1.1	Add a goal model based on the initial do-be-feel list	<a href="#">Hongpei Lu</a>	2022-8-18
2.0	All teams merged the result	<a href="#">Yanan Liu</a> <a href="#">Hongpei Lu</a>	2022-8-19

## Version 2.0

Group cooperation work [here](#)

## Do-Be-Feel-Who List

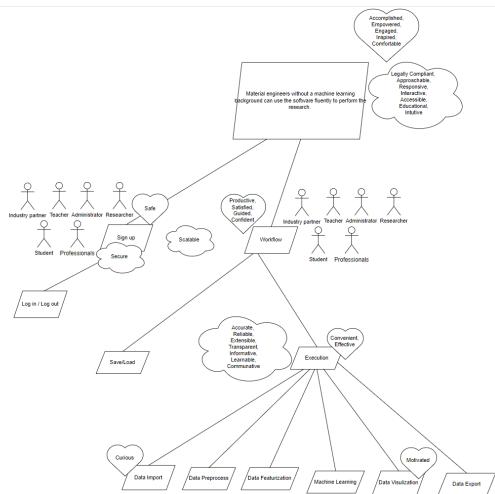
**Overall goal:** Material engineers without a machine learning background can use the software fluently to perform the required data analysis

Who (Roles)	Do(Functional Goal)	Be(Quality Goal)	Feel (Emotion Goal)
Students	Sign up	Accurate	Accomplished
Administrators	Log in/Log out	Approachable	Effective
Professionals	Import Data: Import data files (CSV, XES, Parquet) from local system	Reliable	Satisfied
Industry Partners	Import Data: Drag and drop import of files	Scalable	Productive
Teachers	Import Data: Create working spaces when importing	Extensible	Empowered
Researcher	Data Pre-processing: Overview of the current import data	Transparent (progress, error messages, notebook export...)	Engaged
Code maintainers	Data Pre-processing: Calculate descriptive statistics	Intuitive	Curious
	Data Pre-processing: Reduces noise and eliminates ambiguity	Accessible	Confident
	Data Pre-processing: Consider anonymized data	Secure	Comfortable
	Data Pre-processing: Standardizing data to bring it into the formatting range	Responsive	Guided
	Featurization data: Add multiple composition-based features	Learnable	Motivated
	Featurization data: Add multiple simple density features	Communicative	Convenient
	Machine Learning: Select the machine learning model to be used	Interactive	Safe
	Machine Learning: Define input data and output data: Splitting data into training, test, and validation sets	Legally Compliant	Inspired
	Machine Learning: Determining model features and training the model: Configure and adjust hyperparameters for optimum performance	Educational	
	Machine Learning: Evaluate model performance and establish benchmarks: Evaluate models using validation methods and validation datasets	Informative	
	Machine Learning: Evaluate model performance and establish benchmarks: Continuous measurement and monitoring of model performance		
	Machine Learning: Get model results: The most important features of the current ML model		
	Save / Load workflows		

Export Result: Data Visualization & Comparison - Tabular data & Plotted Graph
Export Result: Pure data table
Export jupyter notebook file
Edit python code directly in the interface
Adding more databases, machine learning methods and plot types
Export Data(input)
Maintain software

## Goal model

Group cooperation work [here](#)



# | User Stories

Assigned to [Rui ZHANG](#)

## Versions

Version ID	Description	Editor	Date
1.0	First version based on initial understanding of the project and first client meeting	Rui ZHANG	2022-8-18
1.1	Added size estimation, priority, and justifications.	Rui ZHANG	2022-8-18
1.2	Supplement the remaining User Stories	Jiahao Ju	2022-8-18
1.3	Changed priority to Moscow priority. Added user story 5.	Rui ZHANG	2022-8-18
1.4	Delete unrelated stories. Add some general stories.	Jiahao Ju	2022-8-19
2.0	Merged user stories from all three teams	Rui ZHANG	2022-8-20
2.1	Reordered the user stories and grouped them by domain. Changed size to estimation of days	Rui ZHANG	2022-8-21
2.2	Regroup Action to new Epic and change original Epic to Domain	Yanan Liu	2022-8-21
2.3	Add epics summary table and explanation for size/priority	Rui ZHANG	2022-9-18

## Version 2.3

Group cooperation work [here](#)

Note: Size was estimated at the end of sprint 1 by technically experienced group members. Size in days indicates workloads. 5-day user stories are large-sized; 3-day user stores are medium-sized; 1 day user stories are small-sized. Priority was set based on client's input.

Epic	Allocation	Total Size
Input Data	RedBack	27
Administration	BoxJelly	31
Machine Learning	BlueRing	16
Data Visualisation	BlueRing	13
Jupyter Notebook	Unallocated - lower priority	20
External Data	Unallocated - lower priority	8

ID	Role	Action	Epic	Domain	Goal	Size (days)	Priority	
30	A s s a	I genera l user want to	be able to view the citations for used featurizers	Input Data	Document ation	so that I could be know more about the source of the featurizer (legally compliant)	1	1 - Must have
32	A s s a	I genera l user want to	browse and select built-in featurizers	Input Data	ML	so that I can discover ways of manipulating my data	1	1 - Must have
34	A s s a	I genera l user want to	browse built-in datasets	Input Data	Server	so that I can discover data to experiment with	1	1 - Must have
19	A s s a	I student want to	quickly browse the Materials available in the database for retrieval and simulations	Input Data	UI	so that I can quickly perform queries.	3	1 - Must have
21	A s s a	I genera l user want to	be able to select datasets from existing databases	Input Data	Data Manipulati on	so that I do not have to worry about how the data is loaded	3	1 - Must have
37	A s s a	I genera l user want to	be able to preview the input data	Input Data	UI	so that I could explore the data	1	2 - Should have
25	A s s a	I genera l user want to	Select specific features from a dataset	Input Data	Data Manipulati on	so that I can improve the precision of my model	3	2 - Should have

13	A s a	Pro user	I want to	add new features	Input Data	ML	so that	they can be reused in the future	5	2 - Should have
28	A s a	genera l user	I want to	be able to reference / view citation for original data sources	Input Data	Document ation	so that	I can retrieve data.	1	3 - Could have
18	A s a	pro user	I want to	be able to apply new featurizers	Input Data	ML	so that	I can create new features	3	3 - Could have
1	A s a	student	I want to	clean and tune data input	Input Data	Data Manipulati on	so that	I have less noise on visualizations.	5	3 - Could have
29	A s a	student	I want to	save project specific data/checkpoints	Administra tion	Server	so that	I can pick up where I left off for specific projects	1	1 - Must have
35	A s a	pro user	I want to	export model selections, parameters, and data flows	Administra tion	Server	so that	I can save my work and share it with others	1	1 - Must have
36	A s a	pro user	I want to	import exported model selections, parameters, and data flows	Administra tion	Server	so that	I can continue work I had previously saved	1	1 - Must have
20	A s a	student	I want to	Create an account using single-sign on, restricted to the *.unimelb.edu.au domain	Administra tion	Server	so that	my research remains private	3	1 - Must have
23	A s a	pro user	I want to	Control job execution	Administra tion	Server	so that	I can start, view progress of, and cancel jobs related to my project	3	1 - Must have
10	A s a	pro user	I want to	be able to opt in to pro-user features	Administra tion	UI	so that	I can access pro user features	5	1 - Must have
38	A s a	pro user	I want to	have my pro user settings persist on each visit	Administra tion	Server	so that	I don't have to reconfigure settings to use the features I need	1	2 - Should have
24	A s a	student	I want to	receive provided hints and guidance for new users	Administra tion	UI	so that	I can quickly learn how to use software	3	2 - Should have
14	A s a	pro user	I want to	easily find and read documentation on the pro features	Administra tion	Document ation	so that	I can use them with ease	5	2 - Should have
17	A s a	pro user	I want to	Be kept informed about job status	Administra tion	Server	so that	I can avoid polling my workspace to check for results	3	3 - Could have
6	A s a	pro user	I want to	have access to more processing power	Administra tion	Server	so that	I can run more complex operations or use more data	5	3 - Could have
31	A s a	genera l user	I want to	able to select a Machine Learning model	Machine Learning	ML	so that	I could use it to train and run the data	1	1 - Must have
33	A s a	genera l user	I want to	browse built-in ML models	Machine Learning	UI	so that	I can discover ways of manipulating my data	1	1 - Must have
39	A s a	user	I want to	be able to select split ratio of data	Machine Learning	Data Manipulati on	so that	to train and test the model	1	2 - Should have
26	A s a	pro user	I want to	have the option to change the hyperparameters used in the machine learning model	Machine Learning	ML	so that	I can fine tune my test results.	3	2 - Should have
15	A s a	pro user	I want to	be able use additional ML models	Machine Learning	ML	so that	I can improve accuracy	5	2 - Should have
7	A s a	pro user	I want to	combine multiple models together	Machine Learning	ML	so that	I can model more complex data manipulations	5	3 - Could have
22	A s a	genera l user	I want to	see clear annotation or explanation of data points and features	Data Visualisati on	UI	so that	I can understand the results of the analysis	3	1 - Must have
8	A s a	student	I want to	use different type of plotting graphs	Data Visualisati on	UI	so that	I have flexibility to visualize data according to my needs.	5	1 - Must have
9	A s a	genera l user	I want to	able to view and plot the results of the model	Data Visualisati on	UI	so that	I could analysis and visualise the effects of the model	5	1 - Must have
12	A s a	student	I want to	export my work to a Jupyter Notebook	Jupyter Notebook	Server	so that	I can extend my work beyond the capability of the application	5	2 - Should have
2	A s a	genera l user	I want to	attach comments to workflow objects	Jupyter Notebook	UI	so that	I can document my work	5	3 - Could have

4	A s a	Pro user	I want to	edit python code on the interface	Jupyter Notebook	UI	so that	I can have control how the ML algorithms works	5	3 - Could have
5	A s a	Pro user	I want to	upload my own script (in python) if possible	Jupyter Notebook	Server	so that	I can extend the tool to support custom models and featurizers	5	3 - Could have
27	A s a	pro user	I want to	be able to access new databases	External Data	Data Manipulati on	so that	I can access additional data	3	2 - Should have
3	A s a	Pro user	I want to	be able to add new datasets in the future	External Data	Data Manipulati on	so that	if there's a new dataset that can be used on a new project, it can be added instantly	5	3 - Could have
11	A s a	student	I want to	analyze the relationship between different features		ML	so that	I can identify which features I need to select for my analysis	5	2 - Should have
16	A s a	genera l user	I want to	add specific materials to the workflow for analysis		Data Manipulati on	so that	compare the performance of the specific material my client or I choose with other material	3	3 - Could have

# | Persona

Assigned to [Xinle Yu](#)

Version	Description	Editor	Date
1.0	First version based on initial understanding of the project and first client meeting	<a href="#">Xinle Yu</a>	2022-8-19
1.1	Round off the sentences and adjust some description	<a href="#">Xinle Yu</a>	2022-8-21
1.2	Combine three team work together	<a href="#">Yanan Liu</a>	2022-8-22
2.0	Adjust technical skills and edit the text for fitting the change of skills	<a href="#">Xinle Yu</a>	2022-9-18
2.1	Combine three team work together	<a href="#">Yanan Liu</a>	2022-9-19

## **Persona 1: Student (By: BoxJelly)**

# Assol Anahita

age: 22

residence: Melbourne

education: Material Engineering

occupation: Student

marital status: Single



*"It's SO time consuming to do material researches and get decent results through just a semester."*

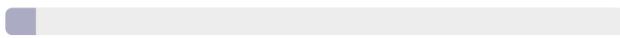
Motivation: As a material engineering graduate student, Assol gets frustrated and demotivated when she can't make sense of the data she has because she doesn't have a tool or sufficient programming/machine learning skills to process the material data. She is also frustrated that she can't use machine learning algorithms to help her engineer new materials even though she is told by her supervisor that this idea works in theory.

## Comfort With Technology

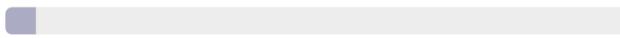
### PROGRAMMING WITH PYTHON



### MACHINE LEARNING



### CLOUD BASED STORAGE



### MATERIAL SCIENCE



## Criteria For Success:

Assol can perform materials data requests/retrievals and accurate materials property predictions supported by Machine Learning technology with easy to follow steps button clicks user interface.

## Needs

- Easy-to-use interface Material science data processing and retrieval application
- A tool to predict property of a material with assistance of Machine Learning technology without prior knowledge of Python and Machine Learning programming

## Wants

- A data mining application that helps her researches
- A better understanding on how Machine Learning can help her to learn more about a material
- Ability to use ML algorithms as a black box
- Freedom to select features on her own terms
- A tool to accelerate research progress

## Values

- Convenience
- Quickness
- Safety
- Understandable

## Fears

- Spends hours without getting anything done because she neither has an adequate tool to do data mining, nor the programming skill to analyse the data herself
- Have to conduct countless experiments to figure out the properties of the materials

- Hard to choose suitable ML algorithms

***Persona 2: Pro User (By: RedBack)***

# Alex

age: 45

residence: Melbourne

education: Masters Degree in Physics

occupation: Materials Engineer

marital status: Divorced without kids



*"There has got to be a better way to do this."*

Motivation : As an experienced Materials Engineer, Alex's job requires him to narrow down candidate materials by performing physical experiments to choose a material which can takes years to do. He needs a tool that can speed up the process by narrowing down candidate materials for experimentation using Machine Learning and simulations.

## Comfort With Technology

### PROGRAMMING WITH PYTHON



### MACHINE LEARNING



### CLOUD BASED STORAGE



### MATERIAL SCIENCE



## Criteria For Success:

Alex can find the right materials efficiently, with accurate results and that matches the client's requirements.

## Needs

- Products to accelerate his workflow
- Access to wide variety of related tools and resources

## Wants

- Suitable models and featurizers for different use cases
- Demonstrate reproducible results to his clients
- Share resources with others
- Refining generated workflow to reuse

## Values

- Extensibility
- Accuracy
- Reliability
- Responsiveness
- Scalability
- Transparency

## Fears

- Tool is too inflexible
- Losing access to progress on his work
- Not being able to verify his results
- Not having support with the tool

***Persona 3: Industry User (By: BlueRing)***

# Gray Zhou

age: 28

residence: Ningde, Fujian, China

education: Master of Material Engineering

occupation: R & D Engineer of Polymer

marital status: Single

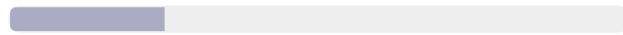


*"It is fantastic to apply a multi-function online tool with ML methods if it is efficient and reliable. Nobody will refuse a tool that can save his time"*

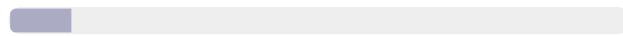
Motivation: Gray Zhou is a R & D Engineer of polymer in a battery factory. His work is searching for better materials for battery production. Gray spends lots of time testing different materials, but some of tests are waste of time because of the poor performance observed. He needs a system that can predict some useful properties of materials so that he can remove samples with low predicted performance and boost the research. His company provides some solutions, but they are awkward and only have limited functions.

## Comfort With Technology

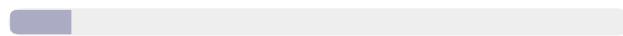
### PROGRAMMING WITH PYTHON



### MACHINE LEARNING



### CLOUD BASED STORAGE



### MATERIAL SCIENCE



## Criteria For Success:

Provide a website or online-tool with quick, visual interface which can help him in daily development of new materials.

A successful product should help him save noticeable time on data processing and provide reliable prediction of properties.

## Needs

- Retrieve and extract required data, process the data with ML methods to get some properties
- Provide graphs which can be modified with interface about predicted properties
- Help finding the material with best predicted properties

## Wants

- Ability to interact with the graph to further compare several materials in detail
- Upload data from his lab for predicting
- Explain what ML method the system applied and how it helps the prediction
- Continue his work on mobile devices without gaps of interaction

## Values

- Easy to get started on both desktop and mobile
- Efficient back-end process
- Abilities to select functions and filter results
- Well organized visualization of interface and

## Fears

- Not enough guidance in the web or tool so him may feel confused to find functions he wants.
- Lacking understanding of what the system does, then reducing the confidence level of his report
- Frequently unavailable.

grids

• Frequency and popularity

# | Prototype

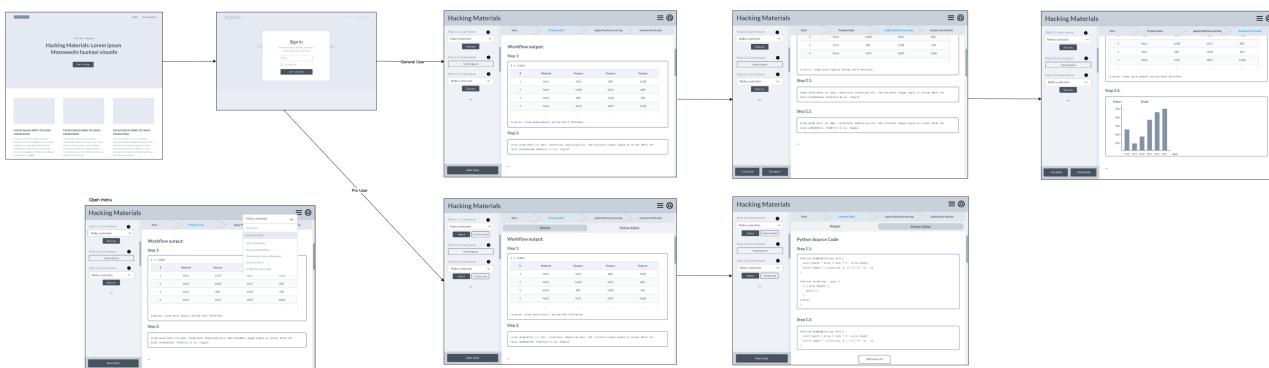
Assigned to [Yanan Liu](#)

## Versions

Version ID	Description	Editor	Date
1.0	First version based on initial understanding of the project and client meeting Made in collaboration with team redback and boxjelly	Yanan Liu Ghina YasharFelipe Leefu Huang Lin	2022-8-20

Group cooperation work, work in [marvel](#)

## Version 1.0



Descriptive Notes recorded by team redback

## Descriptive Notes

- Landing page:
  - Static page with information about the app and project
  - Link to access the app
    - On click, it opens a login modal
    - Once the user is logged in, they're redirected to the app
- Single page app:
  - Top bar:
    - User profile button at the top opens a menu to give the user the option to log out
    - Menu button at the top has options to import or save a workflow, download it in different formats, start over, a link to the documentation and a toggle to enable pro view.
  - General user:
    - The workflow is divided into major and minor steps. Each major step would have its own page. The user can go back and forth between the major steps as needed.
    - Left panel:
      - All the minor steps are numbers and named to guide the user
      - Inputs can be of different types
      - Each step has a tooltip button that would open a modal with guidance information about the step
      - The steps and options in the left panel should always be the same no matter what selections the user made in previous steps. Any step that requires customised inputs would open in a modal.
        - Example 1: Step 3.1 might be "Selecting a plot type". As there is a known, limited list of different plot types, this step may be a drop-down menu that is displayed directly in the left panel.
        - Example 2: Step 3.2 might be customising the selected plot's configuration options. As different plot types may need different configuration options, these options will not be displayed in the panel directly. Instead, the panel will include only a button that says "Configure plot", which would open a modal with the specific options applicable to the selected plot type.

- Pinned buttons at the bottom of the panel: navigate between the different major steps. Last step page may also have a button to download the full workflow.
- Viewing window:
  - At the top of the viewing window, the user can see the progression of major steps with the current step highlighted.
  - The output of each minor step is labelled with the step number and contained inside a box. The output inside the box is the same output produced by running the python code, simply copied over for transparency.
  - The outputs from the previous pages are also always displayed, so it's not just the outputs of the current page.
  - Where a resource with citations is used, the citations will be automatically printed after the output of the step where the resource was selected.
- Pro user:
  - Left panel: has all the same options as a general user, plus additional buttons to configure their own settings as needed
  - Viewing window: the window has 2 tabs:
    - Output: same as the viewing window of the general user
    - Python source code:
      - An editable view of all the code generated by their selections, looks similar to a Jupyter notebook.
      - User can add new cells as desired
      - Brings up the following question: what happens if the user edits the code generated by one of the steps? This may lead to inconsistencies between what is shown in the step's input field and what the code now actually does. This is an implementation decision so is not a major concern right now, but one option that we decided to show in the prototype is that the step's input in the left panel would change to say "Custom" or something similar, indicating that the configuration was changed.

# | Business Case for Cross-Team Collaboration

*Three teams cooperation work original in [here](#), and formatted document can be downloaded here [HA Software Project - Cross-team collaboration - Business case.pdf](#)*

COMP90082 Software Project SM2 2022

## Project Hacking Materials Cross-Team Collaboration Model Proposal

### The current situation

We currently have three teams working with the same client, Dr. Christian Brandl, on the project Hacking Materials ("HA"). The client expressed that he would not be interested in three different versions of the same product, and would prefer we work together to be able to build one more complete final product.

Additionally, it is clear from our discussions with the client that any resulting single product will be difficult to decompose into totally independent components. This means that the deliverables produced by each team will be dependent on those produced by other teams, requiring collaboration on design, development, and project management tasks.

This issue is compounded by the fact that each team has a separate workspace set up by the university, in particular the separate GitHub repositories. This makes it extremely difficult to share resources including source code and development resources and to determine how much progress is being made by each team, and on which tasks.

Our biggest concern is that our teams will be assessed separately. Working on the same product makes us inherently reliant on each other, which could affect the assessment of each team. We understand that this complicates the situation, and have come up with a working agreement that we believe would allow us to deliver the product that the client wants while minimizing the risks to the assessable components of the project.

### Obstacles

We have identified several issues with the way the project is currently being approached, including the following:

- The project scope is larger than what could realistically be accomplished by one student team.
- If the teams were to work completely separately on different parts of the product, it will be difficult to organize ourselves in a way that ensures the needs of all teams are met.
- Finding common availability between 15 people for meetings is difficult. As such, cross-team communication so far has been done mainly through team representatives. This approach is slow, high effort for team representatives (as they need to be aware of everything their teams are doing at all times) and creates an increased risk of information getting lost in transmission.

### Proposed work structure/way of working

The structure we are proposing is as follows:

- The teams would work on the same GitHub repository.
- Each team works as a full-stack team focusing on a particular epic of stories,
- Someone from each team is involved in each technical area of the product (Backend/ Frontend/ Machine Learning).
- The members of the different teams who are within the same technical area cooperate to ensure they follow the same standards and processes. This will help spread some of the cross-team communication load away from the team representatives.
- Each team will need to review the other teams' work to ensure that it doesn't interfere with or affect their own work.
- Ownership of epics will be allocated to teams, but the teams will work together to ensure all high-priority user stories are delivered first. This creates fewer dependencies between teams and allows each team to deliver fully functional components without relying on the others.
- Each team will be responsible for its own Confluence space, but the teams will maintain a similar structure to make navigation easy. Major structural changes will be reported and shared with the other teams during cross-team meetings.
- The teams will share certain deliverables within their Confluence spaces so that the client would not have three different versions of the same document.
- Sprint documents exported from each team's Confluence space will be included within their own folder in the shared GitHub repository.
- The teams will take turns organizing meetings with the client, supervisor, and each other.

### Requested actions

- **The decision from teaching staff:** University administration hasn't made a decision on assessment criteria, or whether the teams can collaborate on this project. This prevents us from adopting a shared working model because we are not sure whether our project submissions will be accepted by the university, or whether we will be otherwise penalized for taking this approach. This was discussed with our supervisor, Mauro, and it was agreed that a response would need to be available by Tuesday 3 pm.
- **Shared GitHub repository:** If our proposal is accepted, we would need a shared GitHub repository so that Dr. Christian Brandl will not have three copies of each deliverable.



# | Development

# Development Process

*Proposed by Red Back team and approved among three teams*

- All tests are required to pass in CI before landing a pull request
- Sprint lifecycle:
  - Sprint Kickoff:
    - Review and re-estimate tasks: user stories get t-shirt size and priority
  - Development:
    - Feature kickoff:
      - Specify test cases and acceptance criteria
      - Tasks are estimated in the number of days to complete using the magic estimation approach
    - Code reviews:
      - Require tests to pass in CI before merging
      - At least one other RedBack member must approve the pull request before it can be merged
      - At least one BoxJelly member working on the same piece of technology must approve the pull request before it can be merged
      - At least one BlueRing member working on the same piece of technology must approve the pull request before it can be merged
      - All test cases and acceptance criteria identified in kickoff must be satisfied
    - Use auto-formatters to maintain code quality
    - Branching
      - Use the format `feature/t-<ticket>` as a feature branch template, where `<ticket>` is the Trello card number
      - `<username>/idea` for scratch/experimenting branches
      - `main` is the main branch
      - We will follow the following guidelines: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow> (we probably only need main, release, scratch, and feature branches)
  - Release management:
    - Deploy as required

# Version Control Guidelines

*Group cooperation work*

## **Code quality guideline**

- Code should be able to run, there's no syntax error
- If test(s) are included in the pull request, they should pass
- All previous tests should still pass with the new changes in the pull request
- A branch should not be put up for a pull request if it has merge conflicts with the main, all conflicts should be resolved before that
- Code should be understandable and contain documentation
- The code should not include global state

## **Code review guideline**

- Has to be reviewed by 1 member from the other 2 teams
- Pull request should have assigned reviewers within 24h once it's no longer DRAFT and should be reviewed (either approve or require changes) within 48h
- Pull requests should include committing messages describing the work you've done and steps you did to verify your work
- Pull request will be reviewed by using the Review Changes button under the Files changed tab
- Pull requests should only be merged/rebased by the creator of the pull request
- Branch that has been successfully merged/rebased to main should be deleted by the creator of the pull request
- Reviewers should follow the work quality guideline to review the code

## **Acceptance criteria definition guideline**

- It should be defined from the user's point of view
- It should contain a list of steps to test the desired functionality

## **Definition of done (for user story)**

- Acceptance criteria should be defined for the user story and pass
- All related code has passed code review and merged to the main

# Test cases

## Versions

Version ID	Description	Editor	Date
1.0	Add test cases for the US 31	Hongpei Lu	2022-9-15

---

## US 31: As a general user, I want to be able to select a Machine Learning model.

### TC 01: Select a model (successful)

<b>Test Type:</b> Functional	<b>Execution Type:</b> Manual
<b>Objective:</b>  Verify if a machine learning model is correctly selected.	
<b>Setup:</b>  The IO stream has been established between the front end and the back end.	
<b>Pre-Condition:</b>  1. The user has inputted the data or selected the data. 2. The user has selected the target feature.	
<b>Notes:</b>  [1] Select the machine learning model in the interface.  Must select a choice in the given place.  Do not select the models that are out of scope in the given choices (Linear regression model and Random forest model).  [2] Submit the information.  * Application starts machine learning using given input dataset and selected feature.	
<b>Time constraint:</b>  Minimum: 10 min  Maximum: 60 min	

### TC 02: Select a model (unsuccessful)

<b>Test Type:</b> Functional	<b>Execution Type:</b> Manual
<b>Objective:</b>  Verify if a machine learning model is correctly selected.	
<b>Setup:</b>  The IO stream has been established between the front end and the back end.	
<b>Pre-Condition:</b>  1. The user has inputted the data or selected the data. 2. The user has selected the target feature.	

**Notes:**

- [1] Try not to select the machine learning model in the interface.  
 [2] Submit the information.  
 \* Application cannot learn without a target selected machine learning model.

**Time constraint:**

Minimum: 10 min  
 Maximum: 60 min

**TC 03: Enter a split ratio of the dataset (successful)**

<b>Test Type:</b> Functional	<b>Execution Type:</b> Manual
<b>Objective:</b>	
Verify if a machine learning model can split a dataset as required.	
<b>Setup:</b>	
The IO stream has been established between the front end and back end.	
<b>Pre-Condition:</b>	
<ol style="list-style-type: none"> <li>1. The user has inputted the data or selected the data.</li> <li>2. The user has selected the target feature.</li> <li>3. The target machine learning method has been selected.</li> </ol>	
<b>Notes:</b>	
<p>[1] Fill in the split ratio in the given place.    * Split ratio cannot be special chars or letters.    * Split ratio must be a number that is bigger or equal to 0 and less or equal to 1.</p>	
<p>[2] Submit the information.    * Application starts machine learning using given input dataset and selected feature.</p>	
<b>Time constraint:</b>	
<p>Minimum: 10 min    Maximum: 60 min</p>	

**TC 04: Enter a split ratio of the dataset (unsuccessful)**

<b>Test Type:</b> Functional	<b>Execution Type:</b> Manual
<b>Objective:</b>	
Verify if a machine learning model can split a dataset as required.	
<b>Setup:</b>	
The IO stream has been established between the front end and the back end.	
<b>Pre-Condition:</b>	
<ol style="list-style-type: none"> <li>1. The user has inputted the data or selected the data.</li> <li>2. The user has selected the target feature.</li> <li>3. The target machine learning method has been selected.</li> </ol>	

**Notes:**

- [1] Try not to fill in the blank.
- [2] Try to fill in the blank using an invalid content-type:
  - [2.1] Try to fill in the blank using special chars.
  - [2.2] Try to fill in the blank using letters.
  - [2.3] Try to fill in the blank using a number that is bigger than 1.
  - [2.4] Try to fill in the blank using a number that less than 0.
- [3] Submit the information.

\*Split ratio cannot be special chars or letters.

\*Split ratio must be a number that is bigger or equal to 0 and less or equal to 1.

**Time constraint:**

Minimum: 10 min

Maximum: 60 min

# Frontend workspace structure proposal

*Proposed by team Red Back and approved among three teams*

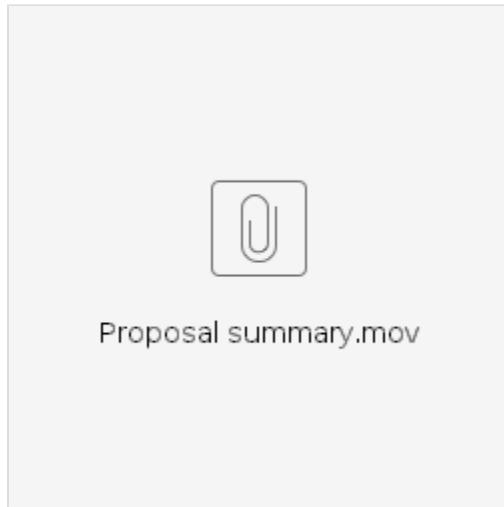
<b>Proposed by</b>	Ghina Yashar
<b>Presented to</b>	All frontend contributors from teams RedBack, BlueRing and BoxJelly
<b>Proposal date</b>	September 1 2022
<b>Status</b>	<span style="border: 1px solid green; padding: 2px;">APPROVED</span> - September 2 2022
<b>Approvers</b>	<ul style="list-style-type: none"><li>■ Mamta Lopes(RedBack): 1/9/2022</li><li>■ Felipe Lin (BoxJelly): 2/9/2022</li><li>■ Rui Zhang (BlueRing): 2/9/2022</li></ul>

## Proposed Structure

**Please note:** All the names used below can be replaced if needed, the focus of this proposal is more on the structure rather than the naming.

## Summary video

If you don't like reading, please watch the video below for a quick overview of the proposed structure. The sample code snippets shown in the video are copied below as well.



## Summary in writing

The structure I'm proposing would follow this rough directory tree:

### Sample directory structure

```

|_assets
|_src
  |_components
    |_exampleComponent
      |_examples.tsx
      |_index.tsx
      |_test.tsx
      |_styled.tsx
    |_dropdownSelectStepType
      |_examples.tsx
      |_index.tsx
      |_test.tsx
      |_styled.tsx
    ...
  |_steps
    |_datasetSelection // (e.g.)
      |_index.tsx
      |_test.tsx
      |_HelpModal
        |_index.tsx
        |_styled.tsx
      ...
    ...
  ...
  |_sections
    |_appHeader
      |_index.tsx
    ...
    |_appBody
      |_InputPanel
        |_index.tsx
      ...
      |_ViewingWindow
        |_index.tsx
      ...
    |_index.tsx
  |_appFooter // amendment suggested by Felipe
  ...
  |_App.tsx
  ...
|_package.json
|_README.md
...

```

The main ideas of this are as follows:

### Sections

By referring to the [low-fidelity prototype](#) created earlier in the project, we divide the main application page into 2 main sections:

- Header: the top bar, which does not need to have context of what stage the user is up to and what's happening at any given point.
- Body: Includes 2 subsections that both need to know which stage the user is at (e.g. "Pre-process data" or "Apply machine learning"):
  - Left-side panel: named in the structure as `InputPanel`. Example code for this panel and how it shows the workflow steps is included in the sample code section below.
  - Main window on the right: named in the structure as `ViewingWindow`

### AMENDMENT - 2 SEPTEMBER 2022

Felipe Lin (HA-BoxJelly) suggests potentially adding an `appFooter` as well. No objections to this so far.

**Hacking Materials** **SECTION: HEADER**  

Step 2.1: Lorem ipsum 

Make a selection 

Execute

Step 2.2: Lorem ipsum 

...  
Lorem ipsum

Step 2.3: Lorem ipsum 

Make a selection 

Execute  
...

Start  Prepare Data  Apply Machine Learning  Analyse the Results

	2	Mat2	1A2B	2012	ABC
3	Mat3	ABC	1A2B	600	
4	Mat4	1010	2009	1A2B	
...	...	...	...	...	...

Citation: Lorem ipsum ndoeuif jhoifgn dofif hofsnfoue

**Step 2.1:**

...  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

**SECTION: BODY**

**Step 2.2:**

...  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

...  
...

< Go back  Go >

**Hacking Materials**  

Step 2.1: Lorem ipsum 

Make a selection 

Execute

Step 2.2: Lorem ipsum 

...  
Lorem ipsum

Step 2.3: Lorem ipsum 

Make a selection 

Execute  
**SUB-  
SECTION:  
INPUT PANEL**

Start  Prepare Data  Apply Machine Learning  Analyse the Results

	2	Mat2	1A2B	2012	ABC
3	Mat3	ABC	1A2B	600	
4	Mat4	1010	2009	1A2B	
...	...	...	...	...	...

Citation: Lorem ipsum ndoeuif jhoifgn dofif hofsnfoue

**Step 2.1:**

...  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

**SUB-SECTION: VIEWING WINDOW**

**Step 2.2:**

...  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

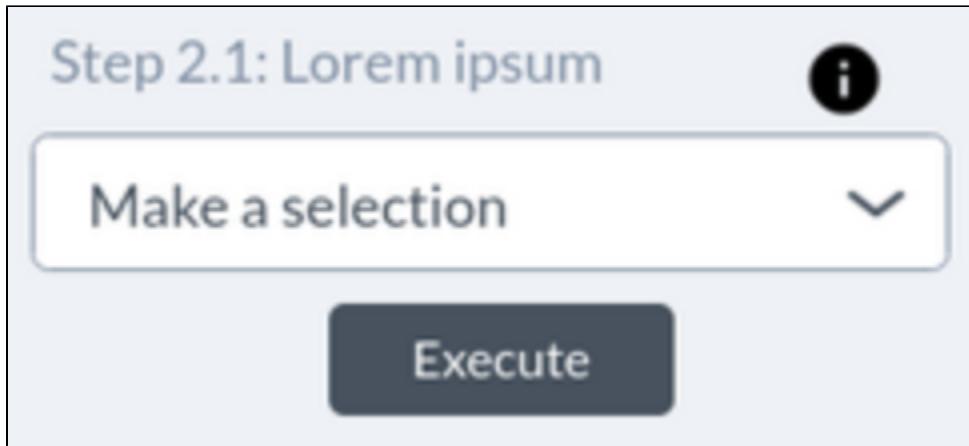
...  
...

< Go back  Go >

## Components

This folder will mainly contain all reusable components, e.g. Button, Tooltip, Modal, etc.

Notably, some of these reusable components would be "step types", e.g. `DropdownSelectStepType`. This example step type refers to the entire object shown below, including a step number, title, tooltip, dropdown list, button and whatever else may be needed. We would create this as a reusable component because many steps have similar requirements, e.g. selecting a dataset and selecting a featurizer should both be dropdown list type steps.



## Steps

The word "steps" in this section refers specifically to the workflow steps that would be shown in the input panel, e.g. Dataset Selection step, Featurizer Selection step, etc.

A separate folder is created for these so that there would be a clear pattern that is easy to follow whenever more steps need to be added. Each step would use a step type component that is imported from the `/components` folder. E.g. the `DatasetSelectionStep` would use the `DropdownSelectStepType`, as shown in the sample code snippet below.

## Sample code snippets

### Sample src/steps/datasetSelection/index.tsx

```
import DropdownSelectStepType from '../../../../../components/dropdownSelectStepType';
import HelpModal from './HelpModal';

...
const DatasetSelectionStep = (props) => {
    ...
    const STEP_KEY = "dataset_selection"

    const options = api_call_here() // calls backend API to get the dataset options

    const onSubmit = selected_value => send_to_backend() // send to backend using api

    return (
        <DropdownSelectStepType
            stepNumber={props.stepNumber}
            title="Select Dataset"
            description="bla bla"
            tooltipContent={HelpModal}
            options={options}
            onSubmit={onSubmit}
        />
    );
};
```

**Sample src/sections/appBody/InputPanel/index.tsx**

```
import DatasetSelectionStep from '../../../../../steps/datasetSelectionStep';
import FeatuirzerSelectionStep from '../../../../../steps/featuirzerSelectionStep';
...

const InputPanel = (props) => {
    ...
    const { stage } = props;

    if (stage === 1) {
        return (
            <div>
                <DatasetSelectionStep
                    stepNumber="1.1"
                    data={data}
                    handleChange={handleChange}
                />
                <FeatuirzerSelectionStep
                    stepNumber="1.1"
                    data={data}
                    handleChange={handleChange}
                />
                ...
            </div>
        );
    } else if (stage === 2) {
        return (
            <div>
                ...
            </div>
        );
    }
};
```

# | Resources

**GitHub:** <https://github.com/COMP90082-2022-SM2/HA-BlueRing>

**Trello:** <https://trello.com/b/Q1AZdIXb/sprint-1>

**Matminer:** [https://nbviewer.org/github/hackingmaterials/matminer\\_examples/blob/main/matminer\\_examples/index.ipynb](https://nbviewer.org/github/hackingmaterials/matminer_examples/blob/main/matminer_examples/index.ipynb)

**LMS Project:** [https://canvas.lms.unimelb.edu.au/courses/126944/pages/hacking-materials-user-interface-code-ha?module\\_item\\_id=3996633](https://canvas.lms.unimelb.edu.au/courses/126944/pages/hacking-materials-user-interface-code-ha?module_item_id=3996633)

**Dr. Christian Brandl information:** <https://findanexpert.unimelb.edu.au/profile/826808-christian-brandl>

**Motivational Model:** <https://momo-staging.eresearch.unimelb.edu.au/dashboard>

# | Contacts

## Teammates

Name	Email address	Student ID
Yanan Liu (Team rep)	<a href="mailto:yananli7@student.unimelb.edu.au">yananli7@student.unimelb.edu.au</a>	1289747
Hongpei Lu	<a href="mailto:hongpeil@student.unimelb.edu.au">hongpeil@student.unimelb.edu.au</a>	1275238
Jiahao Ju	<a href="mailto:jiahaoj1@student.unimelb.edu.au">jiahaoj1@student.unimelb.edu.au</a>	1128182
Xinle Yu	<a href="mailto:xinley@student.unimelb.edu.au">xinley@student.unimelb.edu.au</a>	1221568
Rui Zhang	<a href="mailto:rzzhan2@student.unimelb.edu.au">rzzhan2@student.unimelb.edu.au</a>	1294310

## Client

Name	Email address
Dr. Christian Brandl	<a href="mailto:christian.brandl@unimelb.edu.au">christian.brandl@unimelb.edu.au</a>

## Teaching Staff

Name	Email address
Mauro Mello Jr (supervisor)	<a href="mailto:mauro.mellojr@unimelb.edu.au">mauro.mellojr@unimelb.edu.au</a>
Eduardo Oliveira	<a href="mailto:eduardo.oliveira@unimelb.edu.au">eduardo.oliveira@unimelb.edu.au</a>

## Team RedBack

Name	Email address
Alastair Daivis (Team rep)	<a href="mailto:adaivis@student.unimelb.edu.au">adaivis@student.unimelb.edu.au</a>
Sanjeevani Avasthi	<a href="mailto:savasthi@student.unimelb.edu.au">savasthi@student.unimelb.edu.au</a>
Mamta Lopes	<a href="mailto:mlopes@student.unimelb.edu.au">mlopes@student.unimelb.edu.au</a>
Ghina Yashar	<a href="mailto:gyashar@student.unimelb.edu.au">gyashar@student.unimelb.edu.au</a>
Chunbaixue Yang	<a href="mailto:chunbaixue@student.unimelb.edu.au">chunbaixue@student.unimelb.edu.au</a>
<b>Confluence Page:</b> <a href="#">Home</a>	

## Team BoxJelly

Name	Email address
Zhaoqi Wang (Team rep)	<a href="mailto:zhaoqiw@student.unimelb.edu.au">zhaoqiw@student.unimelb.edu.au</a>
Dara O hEidhin	<a href="mailto:doheidhin@student.unimelb.edu.au">doheidhin@student.unimelb.edu.au</a>
Yaoming Xuan	<a href="mailto:yaomingx@student.unimelb.edu.au">yaomingx@student.unimelb.edu.au</a>
Radhimas Djan	<a href="mailto:djanr@student.unimelb.edu.au">djanr@student.unimelb.edu.au</a>
Felipe Leefu Huang Li	<a href="mailto:fleefuhuangl@student.unimelb.edu.au">fleefuhuangl@student.unimelb.edu.au</a>
<b>Confluence Page:</b> <a href="#">Home</a>	