1. CleverBirds . . . 2
- 1.1 1 - Team Governance & Processes . . . 3
  - 1.1.1 Scrum framework - justifications . . . 3
  - 1.1.2 Scrum Events . . . 3
    - 1.1.2.1 Sprint Planning . . . 4
    - 1.1.2.2 Daily Scrum . . . 5
    - 1.1.2.3 Sprint Review . . . 5
    - 1.1.2.4 Sprint Retrospective . . . 6
    - 1.1.2.5 The Sprint . . . 6
  - 1.1.3 Meeting Notes . . . 7
    - 1.1.3.1 1. Meeting Notes: Kick-off meeting . . . 7
    - 1.1.3.2 2. Meeting Notes: agile methodology & project overview . . . 8
    - 1.1.3.3 3 - Meeting Notes: initial architectural design . . . 8
    - 1.1.3.4 4 - Meeting Notes: working on JIRA & proposal draft . . . 9
    - 1.1.3.5 5 - Meeting Notes: finalizing the proposal document . . . 10
    - 1.1.3.6 6 - Meeting Notes: presentation planning & starting with Confluence . . . 10
    - 1.1.3.7 7 - Meeting Notes: presentation planning . . . 11
    - 1.1.3.8 8 - Meeting Notes: presentation planning . . . 11
    - 1.1.3.9 9 - Meeting Notes: presentation planning . . . 12
    - 1.1.3.10 10 - Meeting Notes: presentation planning with our tutor . . . 12
  - 1.1.4 RACI MATRIX . . . 13
  - 1.1.5 The definition of done (DoD) . . . 13
  - 1.1.6 Performance & Control . . . 14
  - 1.1.7 Gitflow Workflow . . . 15
- 1.2 2 - Project Conception & Life Cycle . . . 16
  - 1.2.1 Project Charter . . . 16
  - 1.2.2 Requirements . . . 17
    - 1.2.2.1 Functional requirements . . . 17
    - 1.2.2.2 Non-functional requirements . . . 17
    - 1.2.2.3 High priority requirements . . . 17
  - 1.2.3 Project Close . . . 19
- 1.3 3 - Planning . . . 19
  - 1.3.1 Product Vision . . . 19
  - 1.3.2 Sprint Plans . . . 20
  - 1.3.3 Project Estimating . . . 20
  - 1.3.4 Risk Management & Quality Control . . . 21
  - 1.3.5 Gantt Chart . . . 21
- 1.4 4 - Architecture & Design . . . 22
  - 1.4.1 Software Design Documentation (SDD) . . . 22
    - 1.4.1.1 Interface Design . . . 22
    - 1.4.1.2 Database Design . . . 26
    - 1.4.1.3 Architectural Design . . . 27
    - 1.4.1.4 Procedural design . . . 27
- 1.5 5 - Build . . . 28
  - 1.5.1 Project structure . . . 28
  - 1.5.2 Getting Started . . . 30
- 1.6 6 - Testing . . . 31
  - 1.6.1 Testing Framework . . . 31
  - 1.6.2 Jira Test Management . . . 32
  - 1.6.3 Feature Testing . . . 32
  - 1.6.4 User Acceptance Tests . . . 34
  - 1.6.5 Release Acceptance Testing . . . 36
  - 1.6.6 Testing Summary . . . 37
- 1.7 7 - Releases . . . 37
- 1.8 8 - Training & Support . . . 37
  - 1.8.1 DBeaver . . . 37
  - 1.8.2 Git flow bug fix . . . 38
  - 1.8.3 GitHub Project Guidelines . . . 39
  - 1.8.4 Flask . . . 39
- 1.9 9 - User Documentation . . . 39

CleverBirds

Documentation overview

This Documentation encompasses all written documents and materials dealing with the software product's development and use.

User guide

Explain how your documentation is organized and how users should navigate this space. Type **/IMAGE** to add a diagram.

Most viewed documents

Add the most viewed documents by typing **/LINK**.

Recently updated

You'll see the 5 most recently updated pages that you and your team create.

- Interface Design
  yesterday at 7:00 pm • contributed by Alice Villar
- Testing Summary
  yesterday at 4:09 pm • contributed by Alice Villar
- CleverBirds
  yesterday at 4:07 pm • contributed by Alice Villar
- User Acceptance Tests

May 20, 2022 • contributed by Alice Villar

Testing Framework
May 20, 2022 • contributed by Alice Villar

1 - Team Governance & Processes

Scrum framework - justifications

Among the various agile methodologies, we have chosen Scrum to maximise resources and time among the popular agile frameworks. As observed in the official Scrum Guide (Schwaber & Sutherland, 2011), successful use of Scrum depends on people becoming more proficient in living five values: Commitment, Focus, Openness, Respect, and Courage. Next, we explain the main reasons for our choice:

1. **Encourages teamwork**: With the division and assignment of roles, Scrum will help us to work successfully on the project.
2. **It is easy to use**: In all stages, Scrum rules are simple and easy to handle.
3. **Saves time:** Scrum will help us to manage tasks in an organized way, which will save time.
4. **Efficiency:** The daily standup meetings provide visibility of one's work to the rest of the team; raise everyone's awareness of who has worked on or is knowledgeable about specific parts of the system; and encourage communications among team members who may not talk to each other regularly. (Chau & Maurer, 2004).
5. **Embraces change:** Scrum embodies a key principle from the Agile Manifesto (Fowler and Highsmith, 2001): "Responding to change over following a plan." It is more important to be flexible than to follow a strict, predefined plan. Scrum enables change throughout rapid product development.

REFERENCES:

Schwaber, K., & Sutherland, J. (2011). The Scrum Guide. Scrum Alliance, 21(1).

Chau, T., & Maurer, F. (2004). Knowledge sharing in agile software teams. In Logic versus approximation (pp. 173-183). Springer, Berlin, Heidelberg.

Educba (2022) What is Scrum?. Available from: https://www.educba.com/what-is-scrum/

Scrum Events

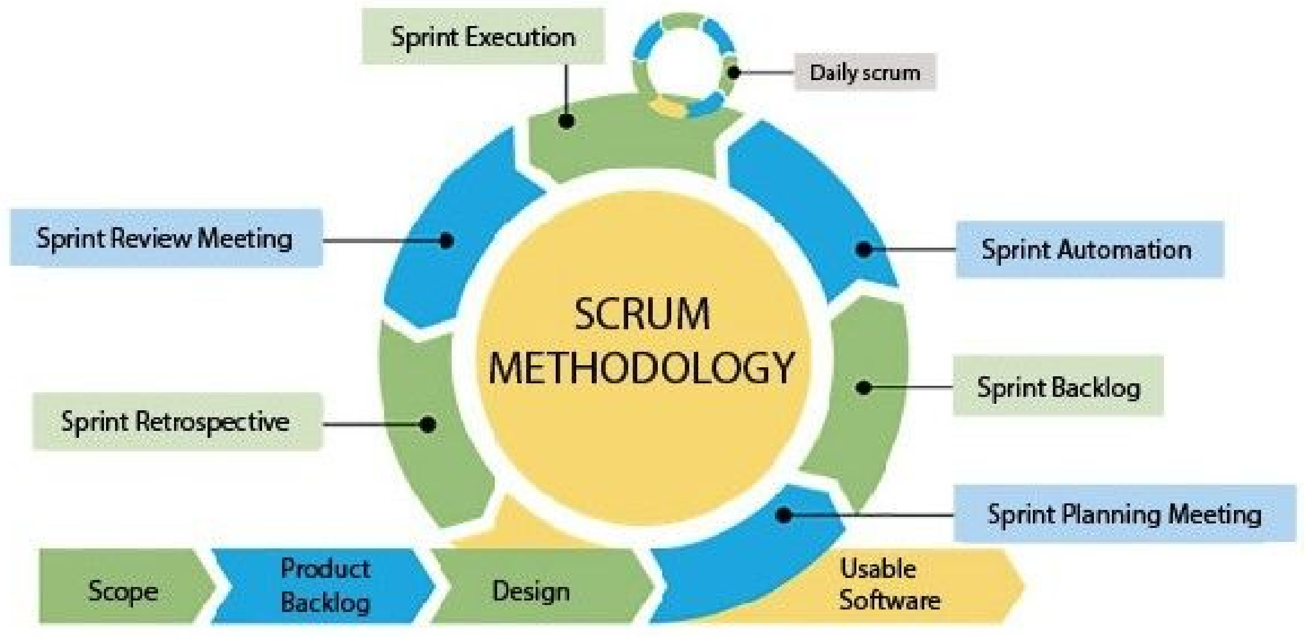
As one of the principles of the Agile Manifesto states:

"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly."

It is with this call for open communication that Scrum encourages us to hold five key events during a Sprint, all intended to help us work efficiently and closely together, as well as to improve our knowledge and become more effective in the future.

There are five main Scrum events:

- Sprint Planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective
- The Sprint



Sprint Planning

CONTENT:

- **1 - What?**
- **2 - When?**
- **3 - Who?**
- **4 - Sprint Planning Questions**

---

1 - What?

Sprint planning is an event in Scrum that kicks off the sprint. A sprint planning meeting is when the team meets to determine which backlog items will be handled in the next sprint.

2 - When?

Our Sprint planning occurs on the first day of a new sprint. The event should occur after the sprint review and retrospective from the previous sprint so that any output from those discussions can be considered when planning for the new sprint.

3 - Who?

Roles	Responsabilites
Project Manager	• Facilitate the ceremony, ensuring that project members share relevant information, openly discuss issues, and hold themselves and each other accountable. • Assure to that the allocated work is being performed as planned. • Answer the Sprint planning questions.
Developer	• Answer the Sprint planning questions.
Technical Writer	• Answer the Sprint planning questions.

- **4 - Sprint Planning Questions**

1. What did we learn at sprint review and sprint retrospective that needs to be considered in our planning conversations today?
2. What is at the top of the product backlog?
3. What do we need to know about these backlog items in order to fully commit to getting them done in the upcoming sprint?
4. How much of the product backlog do we think we can complete in the upcoming sprint?
5. Do we need more clarification on this feature?
6. What is our confidence level regarding this (sprint backlog) as a reasonable plan for this sprint?

7. How is this yet-to-build backlog item similar/relative to other known/completed work we have done in the past? How does this comparison help us estimate the relative size of the new work?
8. What outside help will we need to succeed with our plans?
9. What's the biggest risk that may prevent us from completing this sprint?
10. Who is taking time off during the next sprint? How should we plan differently given the team's (adjusted) capacity?

Daily Scrum

CONTENT:

- **1 - What?**
- **2 - When?**
- **3 - Who?**

---

1 - What?

The purpose of a daily standup meeting is to learn the current progress of every team member that works on Scrum tasks. Daily stand-up meetings align team members around company goals and let them address any short-term challenges that prevent team members from effectively performing their sprint tasks.

2 - When?

- Every day on the Discord app (Monday to Saturday)
- Duration: 5 minutes

3 - Who?

Roles	Responsabilites
Project Manager	• Facilitate the ceremony, ensuring that project members share relevant information, openly discuss issues, and hold themselves and each other accountable. • Assure that the allocated work is being performed as planned. • Answer the following questions: ◦ What did you do yesterday? ◦ What will you do today? ◦ Are there any impediments in your way?
Developer	• Answer the following questions: • What did you do yesterday? • What will you do today? • Are there any impediments in your way?
Technical Writter	• Answer the following questions: • What did you do yesterday? • What will you do today? • Are there any impediments in your way?

Sprint Review

CONTENT:

- **1 - What?**
- **2 - Goals**
- **3 - Topics**
- **4 - Who?**

---

1 - What?

In Agile project management, a sprint review is an informal meeting held at the end of a sprint, in which the Scrum team shows what was accomplished during this period.

2 - Goals

The team gathers to review completed work and determine whether additional changes are required. Other benefits include:

- Maximising responsiveness to customers
- Team building
- Quality assurance

3 - Topics

- When did the sprint begin and finish?

- Who worked on the results?
- What was the goal of the sprint?
- What has been planned?
- What was done and what not?
- What has been added and what was removed from the sprint?
- What risks and problems were discovered?
- Demonstration of developed functionality.
- Feedback from participants.
- Information about the next sprint.

4 - Topics

Roles	Responsibilities
Project Manager	• Present an introduction to our sprint goals. • Arrange the meeting. • Moderator of the meeting • Evidence of feedback.
Developers	• Informs about the sprint status. • Live demonstration of functionality.
Technical Writter	• Answer the Sprint planning questions.

REFERENCES:

https://www.scrumdesk.com/start/manual-for-scrumdesk-start/sprint-review/

Sprint Retrospective

CONTENTS:

1. **What?**
2. **When?**
3. **Purpose**
4. **Who?**
5. **Topics**

---

1. What?

The *Sprint Retrospective* is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

2. When?

According to the Scrum Guide, a sprint retrospective is a meeting held after the sprint review and before the next sprint planning.

3. Goals

The purpose of this meeting is exclusively to collect feedback from the entire team in order to understand which practices worked and which didn't.

4. Who?

Sprint retrospective meetings are intended to serve only the team who executed the sprint.

5. Topics

1. What went right in this sprint?
2. What went wrong in this sprint?
3. What can we commit to in our next sprint?
4. What have you learned from this project?

REFERENCES:

https://easyretro.io/blog/23-questions-to-ask-during-a-sprint-retrospective/

The Sprint

CONTENTS:

1. What?

2. When?

3. Characteristics

1. What?

The Sprint is the heartbeat of Scrum. It is the cadence of the team and should be respected by everyone involved with the Scrum Team.

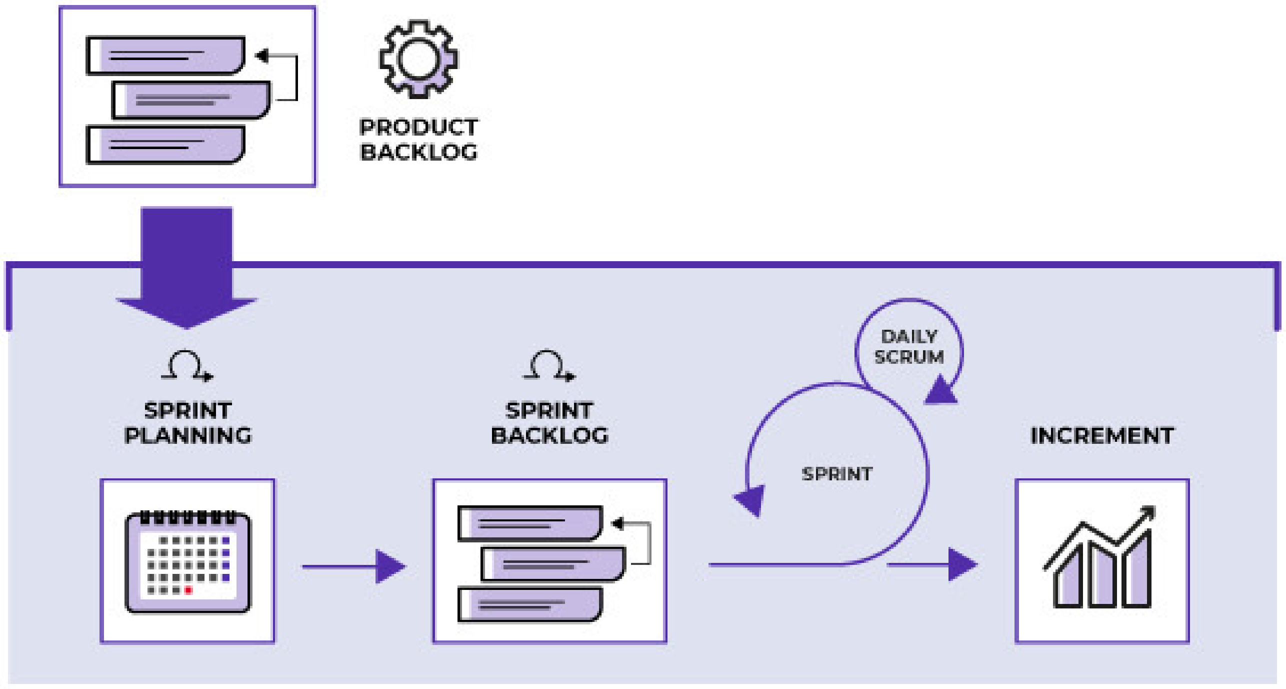
2. When?

A Sprint is usually 2 or 3 weeks. In our Project, each Sprint lasts 2 weeks. The team will attempt to begin, finish, and release work during this time.

3. Characteristics

Some characteristics can include:

- Each Sprint lasts for the same fixed duration (e.g., 2 weeks).
- The next Sprint begins immediately after the previous one.
- The team attempts to begin, work on, complete, and release shippable features.
- The Scrum Team creates a Sprint Goal as the single objective for the Sprint. They then create a Sprint Backlog, which they will use to achieve the Sprint Goal.



REFERENCES:

https://openclassrooms.com/en/courses/4544621-learn-about-agile-project-management-and-scrum/5080956-learn-the-events-of-the-scrum-framework

Meeting Notes

Our team had weekly meetings as an additional opportunity to discuss progress made, what's been slowed down, and what's been stopped. We used DIscord for these meetings and the goal was to allow each member of the team to present their work.

1. Meeting Notes: Kick-off meeting

Date: 12, March 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

- Introductions

- Project background
- Roles
- Collaboration: Talk about JIRA and decide if we will use it in our project.
- Alice will present an initial front-end draft and the team can decide if we will implement a new version of the classical game Flappy Bird.

Notes

- We have selected the requirements that we will send to the other team and we are looking forward to meeting them.
- We discussed how the game in our project could be and came to the conclusion that it will have to be in JavaScript. We will need to redesign the game that we will use.

Action items

- Study agile methodologies so that we can decide if we will use Scrum.
- Do a JIRA course for beginners.
- Do the Agile course from the Youtube channel Simplylearn

Next meeting agenda

1. Decide if we are going to apply Scrum.
2. Start working with JIRA
3. Create a Github repository for our project.

2. Meeting Notes: agile methodology & project overview

Date: 19, March 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

- Discuss agile methodologies and decide which one we will apply.
- Decide if we will implement Clever Birds.

Notes

- Since we are about to work together on Github, everyone should respect and follows best practices. This article presents a brief description of best practices: https://gist.github.com/rsp/057481db4dbd999bb7077f211f53f212
- We decided that we will apply Scrum to our project.
- Having received the requirements from the other team and discussed our project, we decided to implement a new version of Flappy Bird, which will be called Clever Birds.
- Genaro created a repository for our team project and all members are already included.

Action items

- Study Scrum.
- Do a JIRA course for beginners.
- Read and make amendments to our Draft-1, which is available in our shared google drive repository.

Next meeting agenda

- Make a final decision about the roles for each member of the group and discuss how we will rotate roles.
- Present our progress regarding Part 4 of our first draft, which includes the following topics:

· Plan of sprints and phases

· Estimates of implementation time

· Project Estimates Risk Assessment

· Project Acceptance Criteria

· Quality control

· Testing

3 - Meeting Notes: initial architectural design

Date: 26, March 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

- Study Flask Application together.
- Discuss database design, activity diagram, and Flask Structure -http://exploreflask.com/en/latest/organizing.html
- Alice will present the front-end of the project.
- Discuss roles in Scrum and decide our roles.
- Assign tasks to finish our Proposal.
- Talk about Scrum using the material provided by the project manager.

Notes

- Alice presented the front-end of the project and the team agreed with the design.
- Genaro mentioned the fact that the application as it is right now doesn't allow users to change their avatar. We will discuss this in our next meeting when the application will be more advanced.
- Genaro agreed to be responsible for the testing phase.
- Roles: Alice (project manager), Hendrick (Developer), Gennaro (developer), Sharon (technical writer). We still don't know if we will have a "customer service" role in our project.

Action items

- The team will continue to study Scrum and JIRA.
- Hendrick will work on JIRA and the team will follow his work there.
- Alice will revise our first draft considering all the valuable comments from the team and create our Design Document - draft2.

Next meeting agenda

- Hendrick will present JIRA and share his progress (Part 4 or our draft).
- We will start discussing how we will organize our Github repository according to this article which presents best practices: https://gist.github.com/rsp/057481db4dbd999bb7077f211f53f212
- Alice will share the initial Flask app draft and the team will decide if we will carry on the same structure and database design or if Genaro will change it.

4 - Meeting Notes: working on JIRA & proposal draft

Date: 3, April 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

- Hendrick will present JIRA and share his progress (Part 4 or our draft).
- Discuss how we will organize our Github repository according to this article: https://gist.github.com/rsp/057481db4dbd999bb7077f211f53f212
- Alice will share the initial Flask app and the team will decide if we will carry on the same structure and database design or if Genaro will change it.

Notes

The team agreed that the hours for each team member for this project will be 20 hours.

We decided to apply GItflow.

The team still hasn't decided if we will use SQL Server or SQLite, but we shared our opinions and concerns.

Action items

Alice will be responsible for doing the account deletion route and continue working on the proposal draft.

Copolla will be responsible for writing the python code for the game testing.

Sharon will be responsible assist Genaro and Hendrick in the testing phase.

Henrick will be responsible for the Gantt Chart.

The team will revise the Proposal draft to make amendments.

Next meeting agenda

Discuss if we will continue using SQL Server or if we will change to SQLite.

Talk about the development of the Chatbot as a feature of the project.

Each member will share his or her progress with the team.

5 - Meeting Notes: finalizing the proposal document

Date: 17, April 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

- Revise the Proposal draft and share opinions.
- Discuss the tools we will use in the project.
- Discuss Gitflow mistakes that have been made and how we will fix them.

Notes

The team decided to use SQLite.

The team decided what will be our five high-priority requirements.

Our next meeting notes will be done on Confluence.

Alice showed the history of our commits with SourceTree and shared her studies about Gitflow best practices.

Gennaro shared the implementations he did on the code and the team agreed on what we will present in our DEMO.

Action items

Genaro will create activity diagrams for the game.

Hendrick will be responsible for remodeling the Flappy Bird game.

Alice will bring our work together and finalize the Proposal.

Hendrik will work on our project on Jira and Alice will customize our cards.

Next meeting agenda

Discuss the project presentation.

Alice will present an initial draft for our PowerPoint.

Alice will show our Confluence spaces and the team will discuss if we will do our Scrum daily stand-up meetings with Confluence.

Each member of the team will share his or her progress on assigned tasks.

6 - Meeting Notes: presentation planning & starting with Confluence

Date: 23, April 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

Discuss the project presentation.

Alice will present an initial draft for our PowerPoint.

Alice will show our Confluence spaces and the team will discuss if we will do our Scrum daily stand-up meetings with Confluence.

Each member of the team will share his or her progress on assigned tasks.

Notes

- Alice has presented the draft of the PPT for the project presentation
- We have further discussed the content on the PPT of the project presentation
- We finalized Gantt Chart and Budget graph 1.

Action items

- Add some information on the roadmap and test part on JIRA
- Work on the script, PPT and video for the project presentation.
- The whole team must work on our testing plan.
- Do PERT diagram and the Budget graph 2.

Next meeting agenda

- Discuss the script and PPT for the project presentation
- Further, discuss the changes that need to be made on the project (E.g what documents need to create and prepare)

7 - Meeting Notes: presentation planning

Date: 23, April 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

Discuss the project presentation.

Alice will show her progress with Budget charts

- Hendrick will show his progress in the testing phase
- Genaro will show his progress in the testing phase

Notes

- Soon we will be working on the script for the presentation.

Action items

- Alice will continue working on the Budget Charts
- Genaro and Hendrick will continue working on the tests
- The whole team will keep working on the presentation plan

Next meeting agenda

- Team members should present their progress.

8 - Meeting Notes: presentation planning

Date: 30, April 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

Discuss the project presentation.

- Prepare the PowerPoint document

Decide what

Notes

Action items

Next meeting agenda

- Team members should present their progress.

9 - Meeting Notes: presentation planning

Date: 7, May 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

Discuss the project presentation.

- Discuss how the documentation on GitHub should be
- Develop our test plan

Notes

Action items

Next meeting agenda

- Team members should present their progress

10 - Meeting Notes: presentation planning with our tutor

Date: 13, May 2022

Participants

- @ Hendrik @ Alice Villar @ Sharon Wong @ Gennaro Coppola

Agenda

Present our presentation plan to Douglas

- Present Confluence documentation to have feedback from Douglas
- Present how the documentation on GitHub will be
- Ask questions about our test plan and how it should be on the video presentation

Notes

- As our tutor explained, we don't need to give a lot of attention to the ppt document. We should invest our time in the video presentation instead.

Action items

- All the team should finish their tasks as soon as possible so that we can start working on the presentation script.

Next meeting agenda

- Discuss the script of the presentation

RACI MATRIX

RACI-MATRIX-CB.pdf

RACI is an acronym that stands for **responsible, accountable, consulted, and informed**. A RACI chart is a matrix of all the activities or decision-making authorities undertaken in an organization set against all the people or roles.

The definition of done (DoD)

1 - What?

2 - Why?

3 - Clever Birds - DoD

3.1. Project DoD

3.2. Feature DoD

---

1 - What?

The definition of done (DoD) is when all conditions, or acceptance criteria, that a software product must satisfy are met and ready to be accepted by a user, customer, team, or consuming system. We must meet the definition of done to ensure quality.

2 - Why?

Leaving whether or not something is “done” open to interpretation can cause conflict, misunderstandings, and lead to negative user experiences and revenue impacts, which is a good reason to settle on that criteria before the Sprint ever begins.

3 - Clever Birds - DoD

3.1. Project DoD

- The software is in a release-ready state.
- All agreed high priority requirements have been finished.
- The source code has been reviewed by following the four eye principle.
- Code has passed all the unit-tests.
- Code passed integration tests.
- Developer tests passed (performed for each user story and defect).
- User stories passed the UAT.

3.2. Feature DoD

Done on this level may mean it qualifies to add to a release. Not all user stories need to be completed. Rather, it means the feature may be sufficient to satisfy the need. Once accepted, the done feature will contribute to the release velocity. Again, you must meet all of the defined criteria or the feature isn’t done.

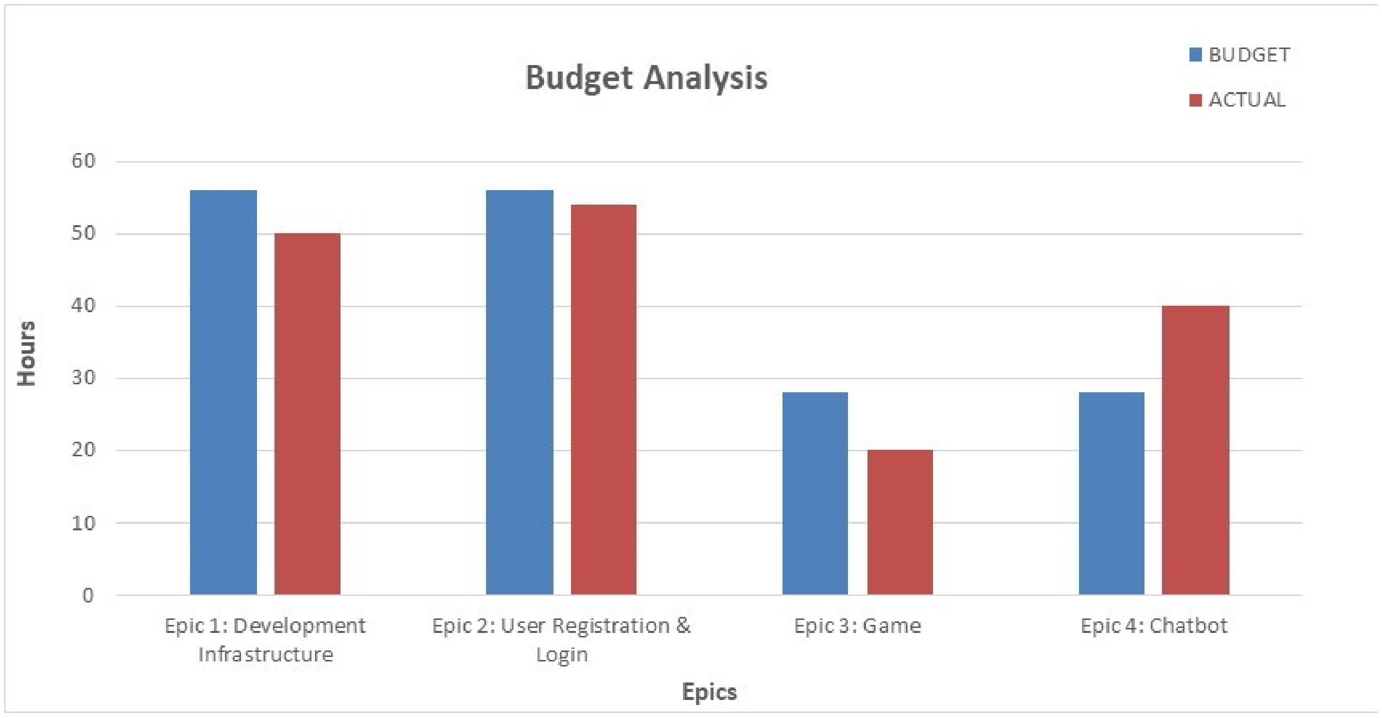
Feature DoD Examples:

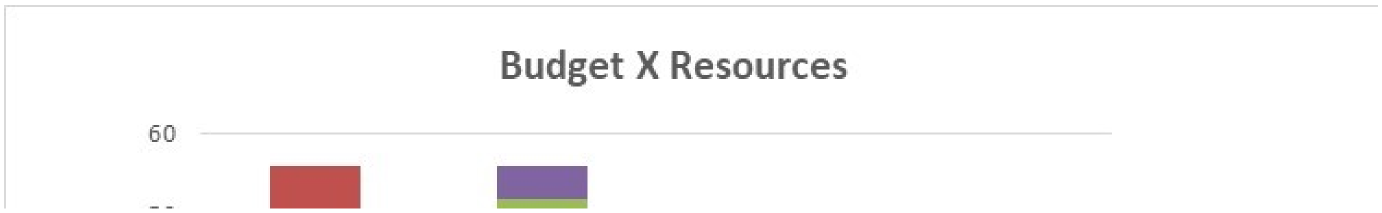
- Integrated into a clean build.
- Manual tests pass.
- Automated tests pass.
- Meets compliance requirements.
- Functionality is documented in the necessary user documentation.

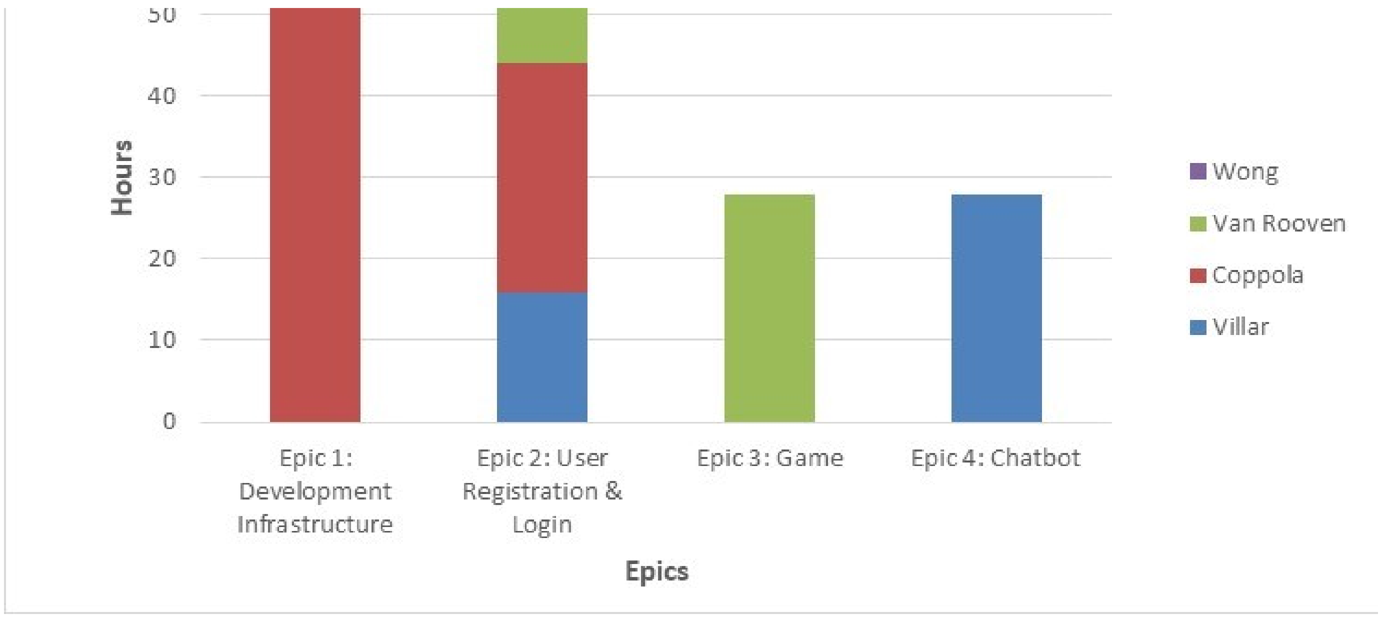
REFERENCES:

https://www.leadingagile.com/2017/02/definition-of-done/#:~:text=The%20definition%20of%20done%20(DoD,of%20done%20to%20ensure%20quality.

Performance & Control







Excel file: https://docs.google.com/spreadsheets/d/1AxbD5a64XZj2MUxReqdDLzualZMfNAzf/edit?usp=sharing&ouid=103356178121864416113&rtpof=true&sd=true

Gitflow Workflow

CONTENT:

- **1 - What is GItflow?**
- **2 - Guidelines for the team**
- **3 - How does it work?**

1 - What is GItflow?

Giflow is an alternative Git branching model that involves the use of feature branches and multiple primary branches. Gitflow has fallen in popularity in favor of trunk-based workflows, which are now considered best practices for modern continuous software development and DevOps practices.

2 - Guidelines for the team

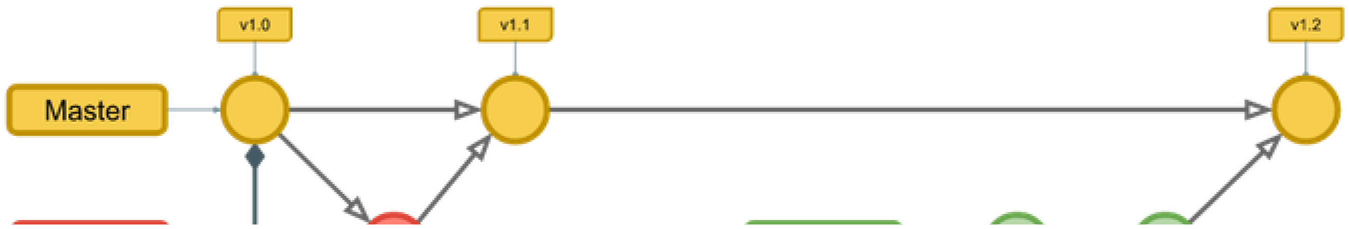
Here is a step by step explanation of how to properly create a new feature:

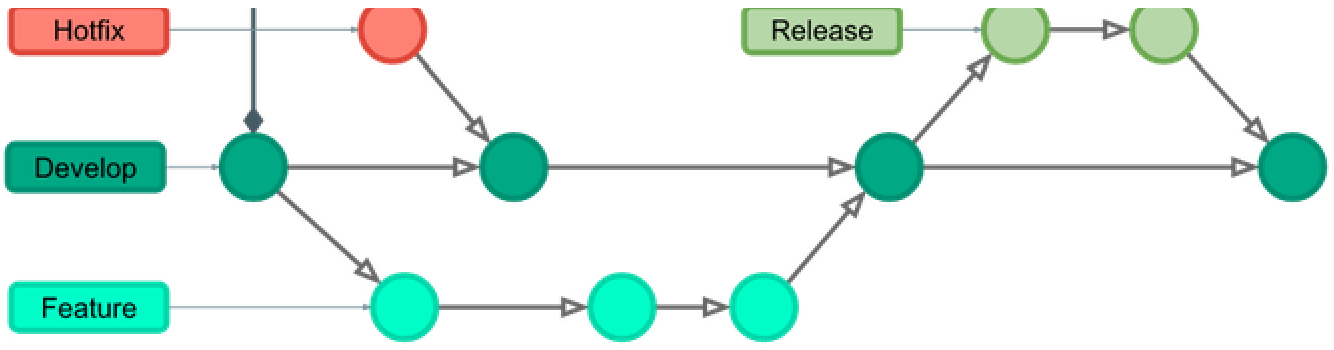
1. Clone the repository (command: git clone 'paste the link') ps.: you only have to do it once. After you have the repository, you just run git pull.
2. Initialize git-flow (command: git flow init) ps.: in the set up you will write "main"
3. Initialize your new feature (command: git flow feature start name-of-your-feature) ps.: no brackets
4. Now you actually do the changes in the code
5. Add your changes (command: git add .)
6. Commit your changes (command: git commit -m "describe what you did")
7. Publish your feature (command: git flow feature publish) ps.: no need to type the name of your feature.
8. Finalize your feature (command: git flow feature finish --keepremote)
9. Upload local repository content to the remote repository (command: git push)

- ALICE'S TUTORIAL VIDEO: https://vimeo.com/705204093/90e8cb8a96
- Please click here to see the git-flow commands I wrote in the terminal: `http://ushortly.com/l31E5`

NOTICE: Normally, the remote feature branch is deleted only if you call `git flow feature finish` with `-F`. However, in our repository, it's being deleted as default. To prevent this, we and the command --keepremote when we finish the feature.

3 - How does it work?





Instead of a single `main/master` branch, this workflow uses two branches to record the history of the project. The `main/master` branch stores the official release history, and the `develop` branch serves as an integration branch for features. It's also convenient to tag all commits in the `mai`
 branch with a version number.

Each new feature should reside in its own branch, which can be pushed to the central repository for backup/collaboration. But, instead of branching off of `main`, feature branches use `develop` as their parent branch. When a feature is complete, it gets merged back into `develop` . Features should never interact directly with `main`.

Note that `feature` branches combined with the `develop` branch is, for all intents and purposes, the Feature Branch Workflow. But, the Gitflow workflow doesn't stop there.

`Feature` branches are generally created off to the latest `develop` branch.

References:

https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow

https://danielkummer.github.io/git-flow-cheatsheet/

https://www.youtube.com/watch?v=R5QJ0FsD12Q

https://git.logikum.hu/flow/feature

Introduction-of-Continuous-Delivery-in-Multi-Customer-Project-Courses.pdf (researchgate.net)

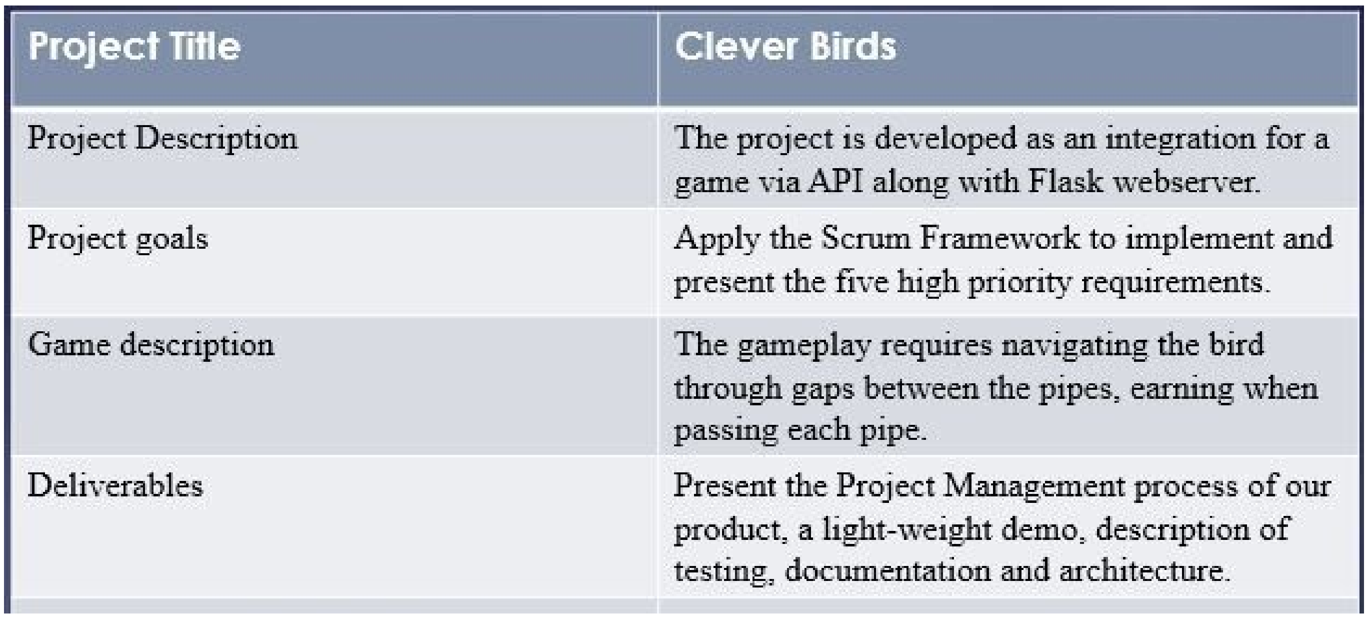
2 - Project Conception & Life Cycle

In the initiation phase, we defined our project's goals, requirements, scope, and roles of each member. Main documents include:

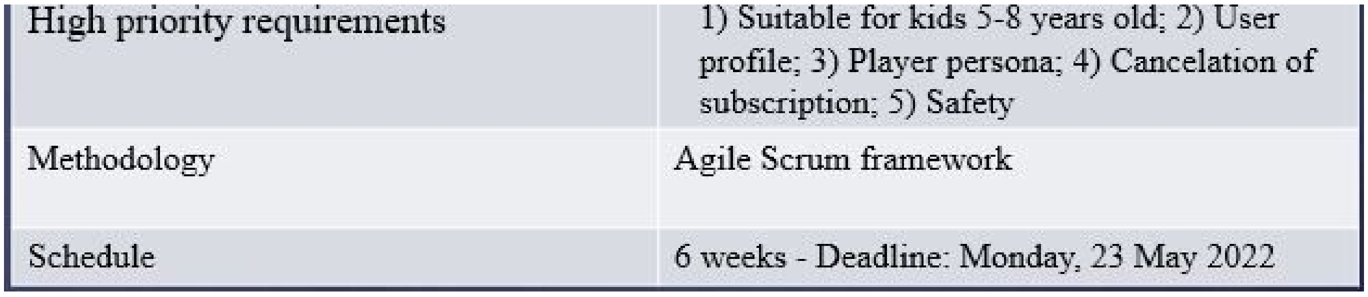
- Project proposal
- Project charter
- RACI chart

Finally, the project was kicked off and started.

Project Charter

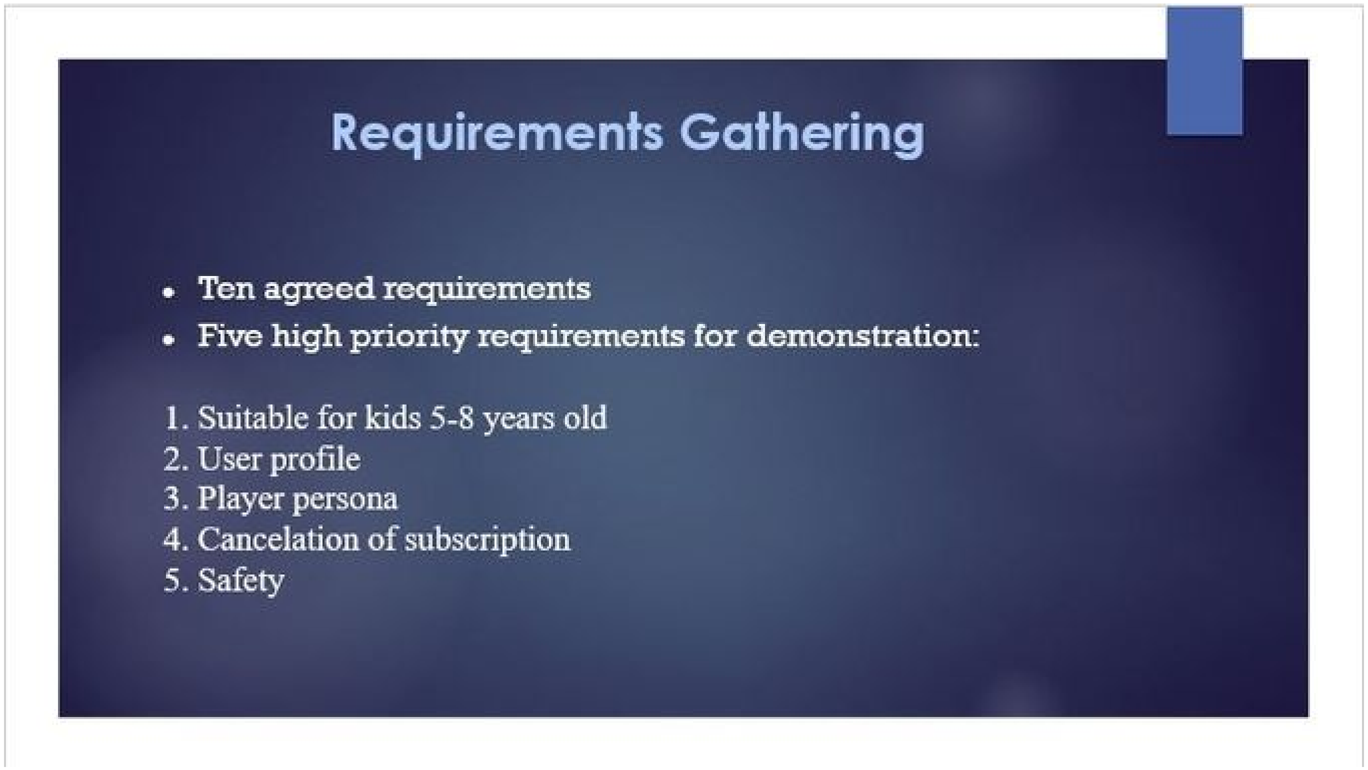


Project Title	Clever Birds
Project Description	The project is developed as an integration for a game via API along with Flask webserver.
Project goals	Apply the Scrum Framework to implement and present the five high priority requirements.
Game description	The gameplay requires navigating the bird through gaps between the pipes, earning when passing each pipe.
Deliverables	Present the Project Management process of our product, a light-weight demo, description of testing, documentation and architecture.



High priority requirements	1) Suitable for kids 5-8 years old; 2) User profile; 3) Player persona; 4) Cancelation of subscription; 5) Safety
Methodology	Agile Scrum framework
Schedule	6 weeks - Deadline: Monday, 23 May 2022

Requirements



Requirements Gathering

- Ten agreed requirements
- Five high priority requirements for demonstration:

1. Suitable for kids 5-8 years old
2. User profile
3. Player persona
4. Cancelation of subscription
5. Safety

Functional requirements

- **a1) Fun**: Clever Birds is fun and simple. According to Rosewater (2011), the real way to determine if a game is fun is to ask the players if they would play again. We will do this through our chatbot Tweety in a customer satisfaction survey.
- **a2) User Profile**: Users can create a user profile.
- **a3) Suitable for kids 5-8 years old**: Clever Birds is very simple. After signing in, the user can see a green button on the top right ("PLAY NOW!") on all screens. By pressing the button, the user starts playing.
- **a4) Unique selling point**: A chatbot to talk to parents about the skills their children can gain.
- **a5) Player persona**: The application allows users to choose between three avatars.
- **a6) Cancelation of subscription**: On the page "Manage Account", users can delete their account, which will erase their data from our database.
- **a7) Safety**: The chatbot "Tweety" teaches parents how to protect their children by discussing video games' health problems and how to solve them.

Non-functional requirements

- **b1) UI for single-hand us**e: In the Clever Birds game, the player pilots the bird past the pipes by clicking with the mouse.
- **b2) Most efficient data storage**: Our application stores data efficiently as explained in section X.
- **b3) Most efficient data management and search**: In our application data can be managed efficiently, as explained in Section X.

High priority requirements

From the agreed ten requirements received, we have chosen five as high priority due to their importance to the overall product functionality and design.

Requirement	**Justification**	**Gherkin Language**
a3	The game should be easy to understand and the user’s path to playing the game very simple.	**Scenario: play Clever Birds game** Given the user is registered and logged in When the user clicks on the button “PLAY NOW!”, which is present is all the website pages Then the game should start
a5	One of the key features of Clever Birds is the ability for a user to create a user profile.	**Scenario: create user profile** Given the user is on the homepageWhen the user clicks on the button “Create an Account” Then the “Create your profile” page should display **Scenario: navigate to avatar selection page** Given the user is on the “Create your profile” page When the user successfully fills out the required fields And the user clicks on “Next Step” Then the “Choose Your Avatar” page should display **Scenario: avatar selection**Given the user is on the “Choose Your Avatar” page. When the user selects an avatar Then a page showing that the account was successfully created should display.
a7	Another key feature setting the application apart from Flappy Birds is a safety feature informing parents about health problems related to video games.	**Scenario: access the chatbot** Given: application is open When: the user clicks on “Talk to Tweety” Then a page showing the chat dialog should display.
a6	In order to comply with the GDPR, we allow users to delete their account at any time and their data will be deleted from our database.	**Scenario: open menu** Given: the user is logged in When: the user clicks on the menu icon Then: the menu should open **Scenario navigate to “Manage Account” page** Given the menu is open When the user clicks on “Manage Account” Then the “Manage Account” page should display **Scenario: Delete account** Given the “Manage Account” page is open When the user clicks on “Delete Account” Then the account and associated data is completely removed from the Clever Birds database.
a2		**Scenario: create user profile**

	User profiles play a vital role in user experience as it provides a collection of information associated with users.	Given the user is on the homepageWhen the user clicks on the button “Create an Account” Then the “Create your profile” page should display

Project Close

In project management, project closure is a formal written assessment of a project. It documents all phases of project management into one digestible report. Through introspection, a project manager learns what worked and what didn't.

The Project Post-Mortem Retrospective

What we liked

- Team spirit and a positive working atmosphere.
- Diverse skillsets keep us engaged and excited to complete tasks.
- Daily standup scrum encouraged positive team building and improved collaboration.
- Working with Confluence was great because makes the work more visible. We were able to bring all information into one place and this improved the project a lot.

What we missed

We didn't have time to work on more Key performance indicators (KPIs) in project management.

What I learnt

- We learnt to use JIRA & Confluence, which eliminated the need for shared drives and file folders.
- We learnt how to operate in a team with Gitflow Workflow.
- Practice strategic communication, problem-solving.
- We learnt how to create effective project documentation and artifacts throughout the various phases of a project

For next time

- We would definitely make time to work on more Key performance indicators (KPIs) in project management because they help focus attention on what matters most.

Appreciations

- We are grateful to our tutor for the extra time he has given us in order to make us achieve what we wanted.

3 - Planning

- Product Vision
- Sprint Plans
- Project Estimating
- Risk Management & Control
- Gantt Chart

Product Vision

Problems & Solutions

1 - Kids and Video Games: Health and Safety Issues

- **Solution**: Some of the main dangers of video games are cyberbullying, privacy problems, personal information on consoles, computers, and devices, webcam worries, online predators, and hidden Fees (Kapersky, 2022) Our project introduces a chatbot named “Tweety”. It will be able to explain to parents how to protect children their children and enhance the gaming experience by discussing health problems related to video games and how to solve them. Tweety’s scripts will be created using literature reviews about health problems related to the addiction to video game playing, such as the following: Al-Bayed & Naser, 2017; Baranowski, et al., 2012, LeBlanc et. al, 2013.

2 - Complexity

- **Solution**: Kids from 5 years old can use Clever Birds without having to ask for help. It is very simple to sign in and by pressing a button on the top right of the screen ("PLAY NOW") the user starts playing. Every screen in the application has that button.

3 - Gameplay engagement challenges

- **Solution**: On the page "Choose your avatar", the application allows users to choose one of the following avatars: Daisy, Alfredo, and Birdy. This feature is key to driving customer engagement.

4 - Comply with GDPR

- **Solution**: The project follows all privacy and security regulations, including those specified by the GDPR. Thus, personal data collected from users is securely stored and deleted upon request.

5 - Focus and organization of user's information

- **Solution**: Clever Birds application allows users to visualize their user profiles. This plays a vital role in user experience as it provides a collection of information associated with them.

REFERENCES:

Kapersky, 2022. How to Protect Your Child from the Top 7 Dangers of Online Gaming. Available from: https://usa.kaspersky.com/resource-center/threats/top-7-online-gaming-dangers-facing-kids

Al-Bayed, M. H., & Naser, S. S. A. (2017). An intelligent tutoring system for health problems related to addiction of video game playing. *International Journal of Advanced Scientific Research*, *2*(1).

LeBlanc, A. G., Chaput, J. P., McFarlane, A., Colley, R. C., Thivel, D., Biddle, S. J., ... & Tremblay, M. S. (2013). Active video games and health indicators in children and youth: a systematic review. *PloS one*, *8*(6), e65351.

Baranowski, T., Abdelsamad, D., Baranowski, J., O'Connor, T. M., Thompson, D., Barnett, A., ... & Chen, T. A. (2012). Impact of an active video game on healthy children's physical activity. *Pediatrics*, *129*(3), e636-e642.

Sprint Plans

Sprint 1

CONTENTS:

- **Duration:** 03/04 - 18/04
- **Objective:** Development infrastructure
- **Who will participate:** Coppola

Sprint 2

CONTENTS:

- **Duration:** 17/04 - 02/05
- **Objective:** User Registration & Login
- **Who will participate:** Coppola, Rooven, Villar, Wong

Sprint 3

CONTENTS:

- **Duration:** 04/05 - 17/05
- **Objective:** Game & Chatbot
- **Who will participate:** Rooven and Villar

Project Estimating

The following table presents our Epic estimations:

Epic	**Story Points (1 SP = 4 Hours)**
CB-37: Development Infrastructure	14
CB-34: User Registration & Login	14
CB-60: Game	7
CB-66: Chatbot	7

- 1 Story Point = 4 hours.
- In our Ghantt Chart, 1 day corresponds to 4 hours of work.
- Thus, we have 28 hours per week (4x7=28).
- Each Sprint has two weeks and therefore 56 hours of work.
- Each Sprint has 14 Story Points, so we have 56 hours (14x4=56).

Planning Poker

We used the *Planning Poker* technique in our sprint planning meetings to get team members to come to a consensus on how many points to assign to a user story.

Planning poker, also known as “scrum poker” and “pointing poker”, is a gamified technique that development teams use to guess the effort of project management tasks. These estimations are based on the entire group's input and consensus, making them more engaging and accurate than other methods.

Risk Management & Quality Control

Risk Management

Risk management is the discipline of identifying, monitoring and limiting risks. Agile methodology has an iterative approach that enables continuous attention to risks and the risks can be reduced by different practices like continuous software integration and early validation.

In our project, the risk management involves:

- Identifying the risk
- Analyzing each risk to determine its exposure (severity of impact)
- Prioritizing the identified risks based on their exposure
- Creating action plans (responses) to deal with the high-priority risks
- Continuous monitoring and follow-up to ensure that your action plans are mitigating the risks

Quality Control

While risk management revolves around project objectives and focuses on the probability of achieving success criteria defined at the beginning of the project; quality management goals focus on improving, developing, and testing processes with the aim of preventing defects.

Quality Assurance is the set of activities aiming to build quality into the final product.

It involves every participant in the lifecycle of a product's development, at any level (regardless of seniority or the level he/she is in the organizational structure). As long as someone can influence the development of a product and/or service, he/she is responsible for the quality of the result.

Quality assurance activities can be split into two categories. First aims to prevent the defects that would arise before the development cycle ends, by defining a standard way of working, checkpoints and reviews. Measurements of various KPIs, analysis of the results, taking corrective and improvement actions, as well as regular assessments of the way of working (at all levels), are part of this process. The second category focuses on defect detection in the completed product and is better known as Quality Control.

REFERENCES:

https://www.todaysoftmag.com/article/1367/a-simple-approach-for-risk-management-in-scrum

https://www.apm.org.uk/blog/successful-quality-management-requires-expert-risk-management/

https://www.todaysoftmag.com/article/1355/quality-assurance-in-agile-scrum-environment

Gantt Chart

A Gantt chart is a graphical representation of activity against time, illustrating a project plan. It typically includes two sections: the left side outlines a list of tasks, while the right side has a timeline with schedule bars that visualize work.

1 day = 4 hours

Gantt Chart CB.pdf

4 - Architecture & Design

Software Design Documentation (SDD)

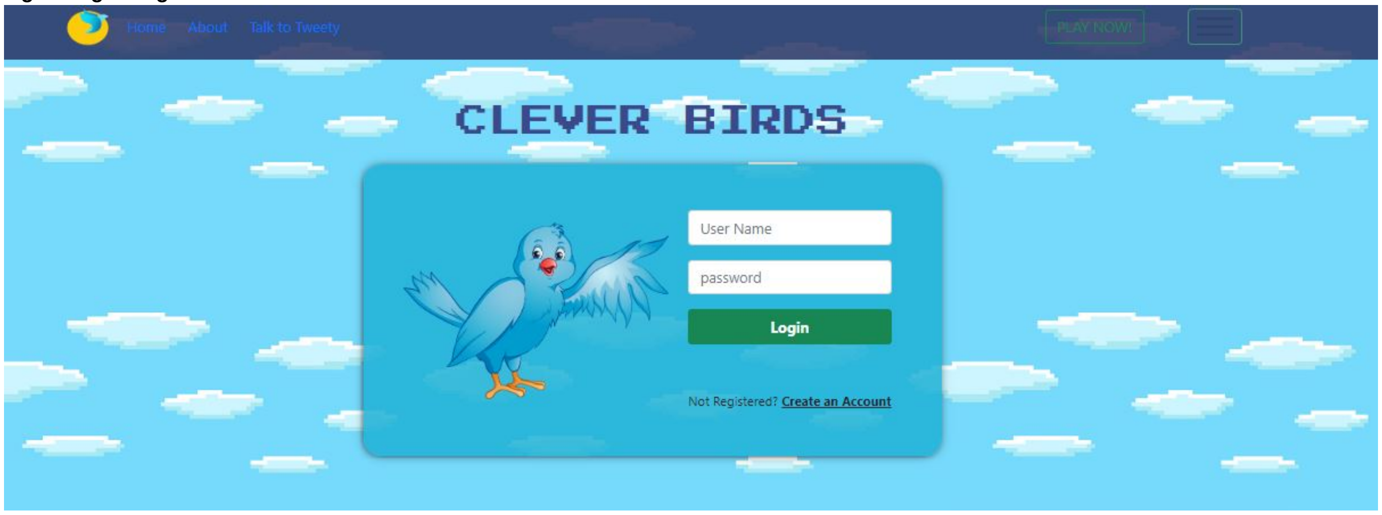
The Software Design Document (SDD) is a comprehensive software design model consisting of four distinct but interrelated activities: interface design, database design, architectural design and procedural design.

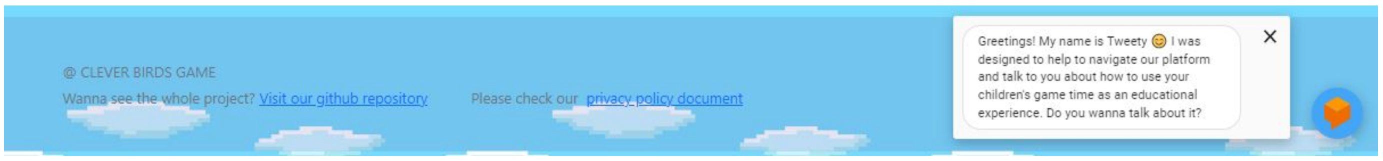
Interface Design

This table shows the user interface design (UI) plan for Clever Birds, followed by a print screen of each webpage.

Login/Register	Fig. 1
Create account Step 1 - User Profile **Create account Step 2 - Choose your avatar**	Fig. 2 Fig. 3
Account successfully created	Fig. 4
User Dashboard	Fig. 5
Manage account	Fig. 6
Talking to Chatbot Tweety	Fig. 7
Play the game	Fig. 8
About	Fig. 9
Privacy Policy	Fig. 10

Fig.1: Login/Register

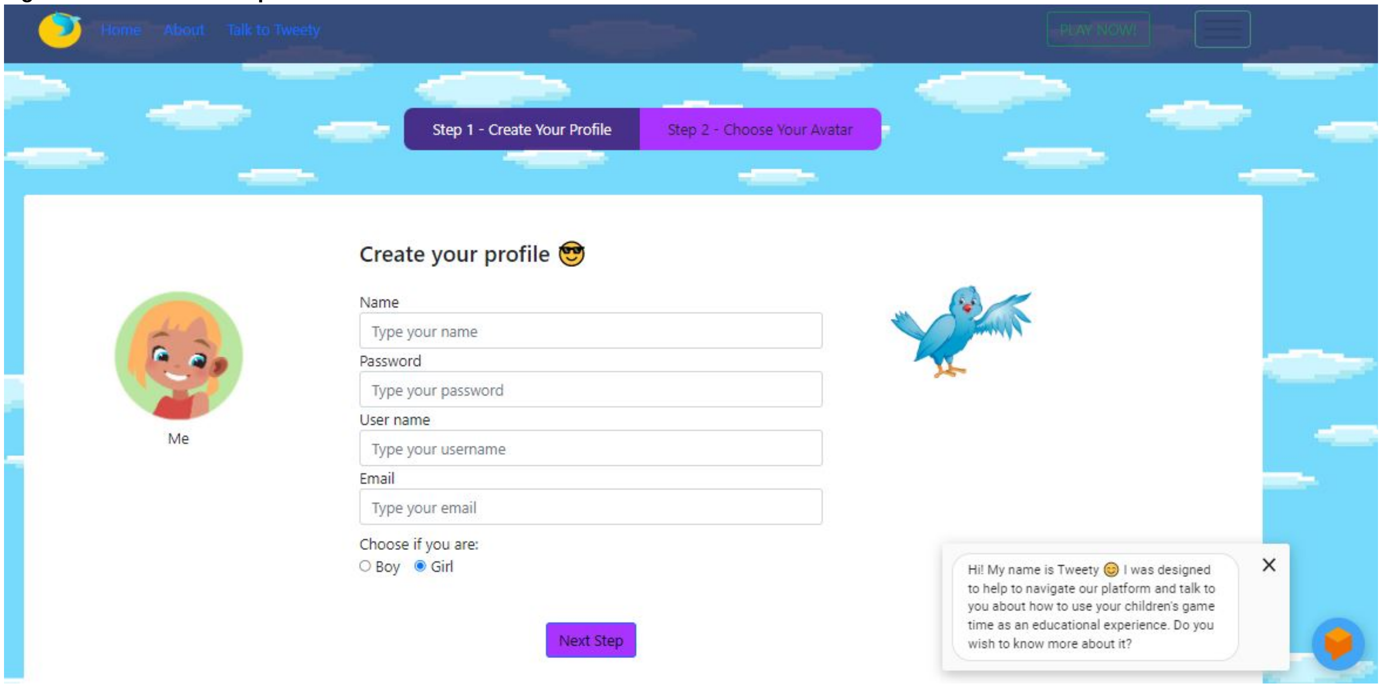



@ CLEVER BIRDS GAME

Wanna see the whole project? Visit our github repository   Please check our privacy policy document

Greetings! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wanna talk about it?

Fig.2: Create account Step 1 - User Profile



Create your profile 😎

Name: Type your name

Password: Type your password

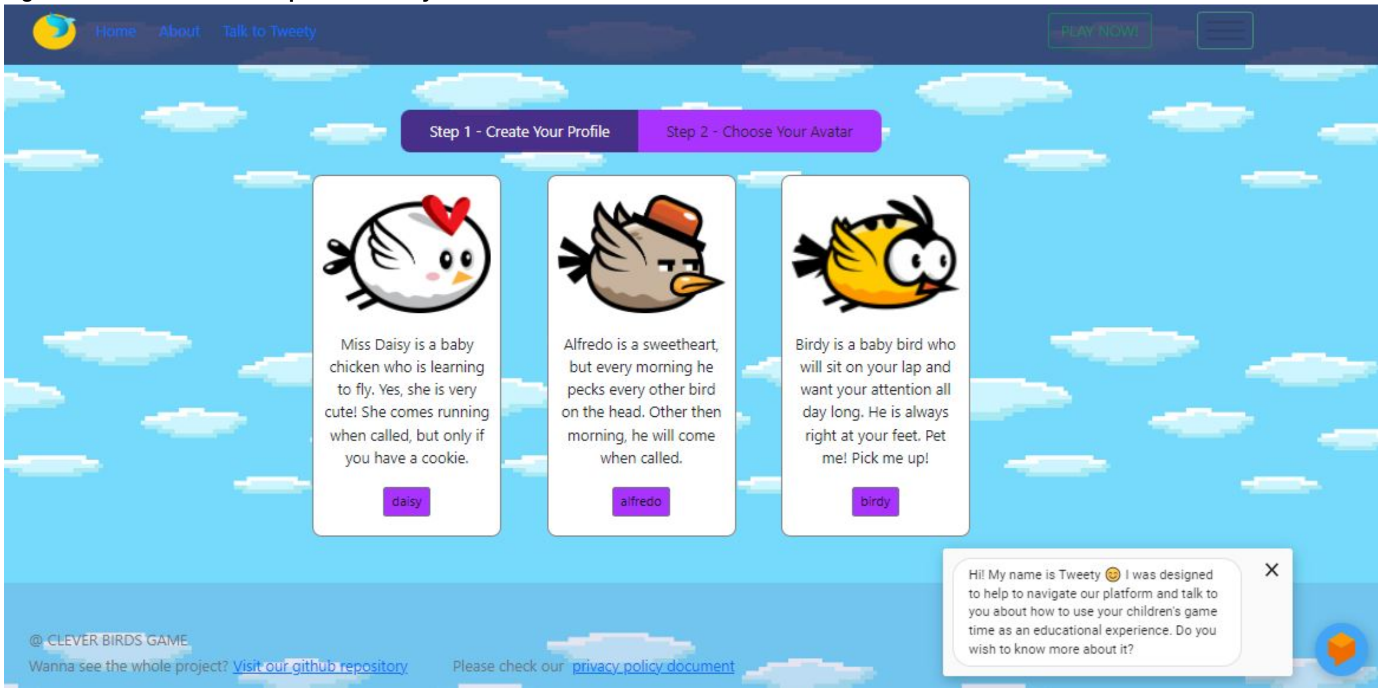
User name: Type your username

Email: Type your email

Choose if you are:
Boy [✓ Girl]

Hi! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wish to know more about it?

Fig.3: Create an account Step 2 - Choose your avatar



Miss Daisy is a baby chicken who is learning to fly. Yes, she is very cute! She comes running when called, but only if you have a cookie.

Alfredo is a sweetheart, but every morning he pecks every other bird on the head. Other then morning, he will come when called.

Birdy is a baby bird who will sit on your lap and want your attention all day long. He is always right at your feet. Pet me! Pick me up!

Hi! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wish to know more about it?

@ CLEVER BIRDS GAME

Wanna see the whole project? Visit our github repository   Please check our privacy policy document

Fig.4: Account successfully created

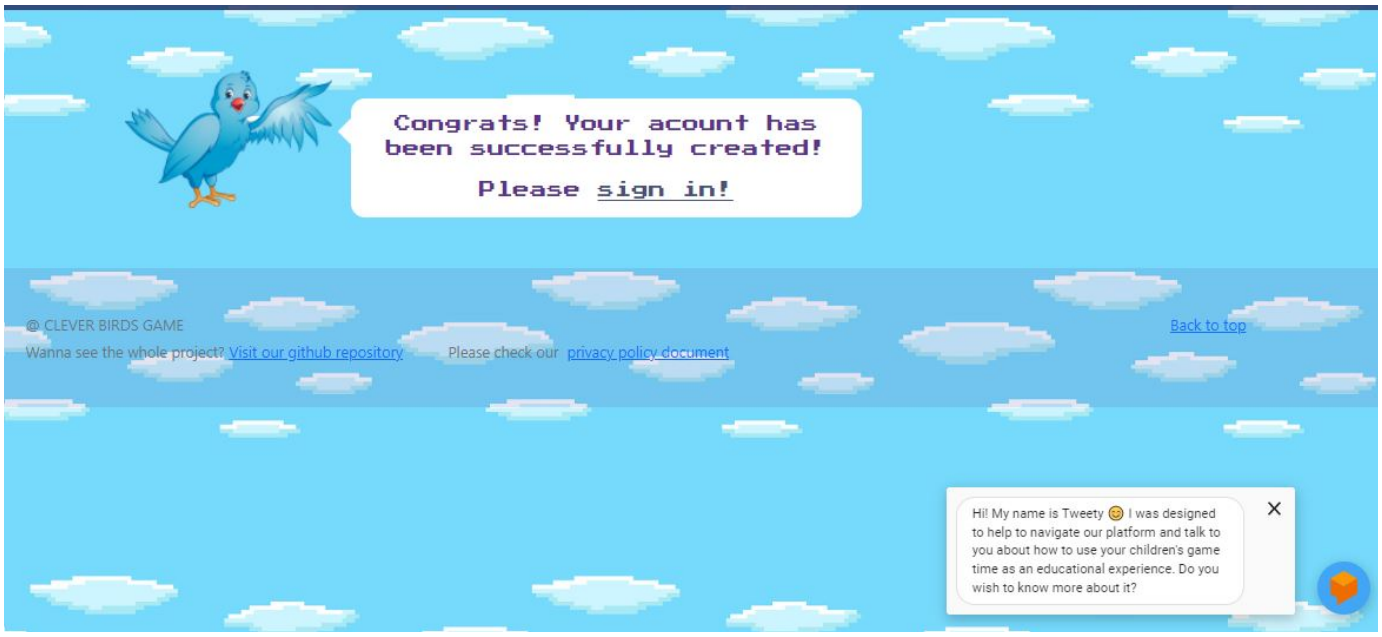
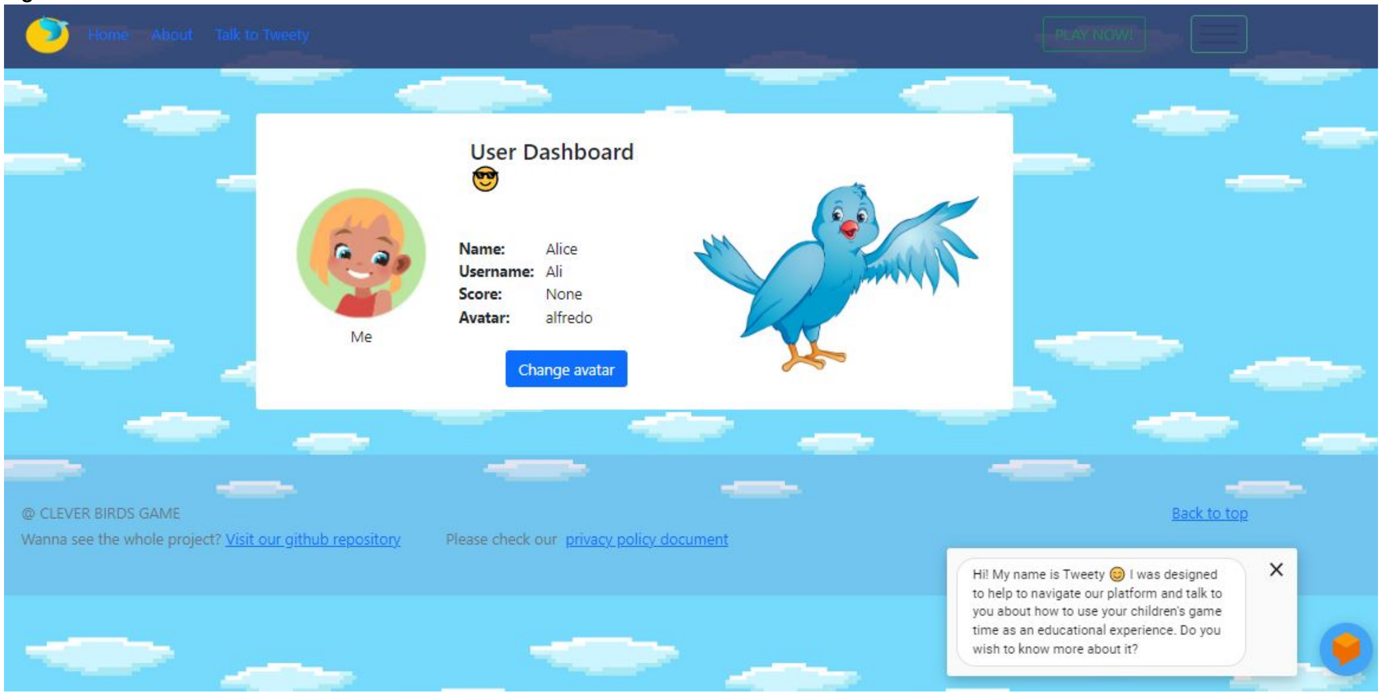
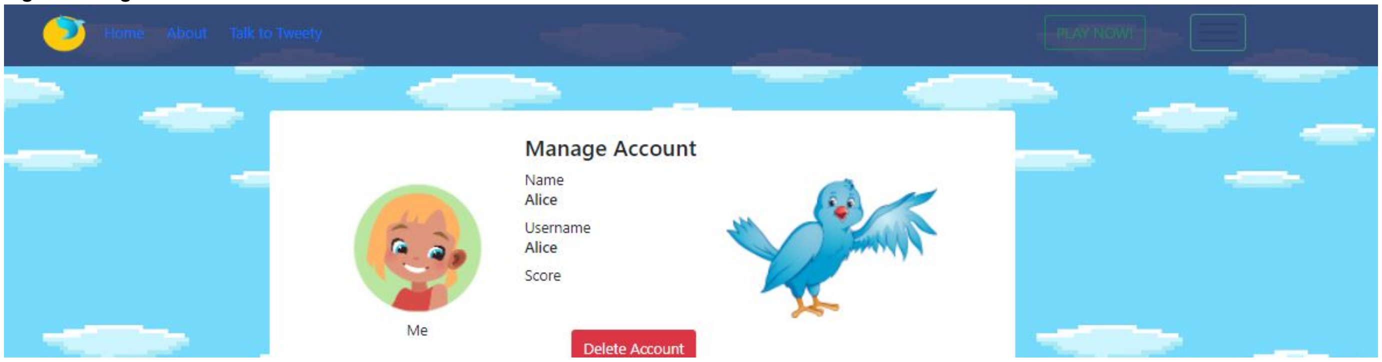

Congrats! Your acount has been successfully created!
Please sign in!
@ CLEVER BIRDS GAME
Wanna see the whole project? Visit our github repository
Please check our privacy policy document
Back to top
Hi! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wish to know more about it?

Fig.5: User dashboard


Home
About
Talk to Tweety
PLAY NOW!
User Dashboard 😎
Me
Change avatar
@ CLEVER BIRDS GAME
Wanna see the whole project? Visit our github repository
Please check our privacy policy document
Back to top
Hi! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wish to know more about it?

Name:	Alice
Username:	Ali
Score:	None
Avatar:	alfredo

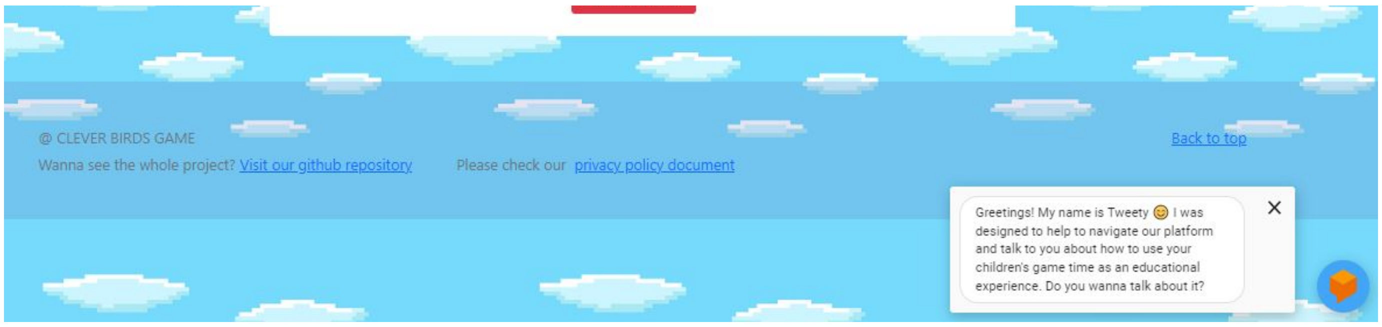
Fig.6: Manage account



Name
Alice

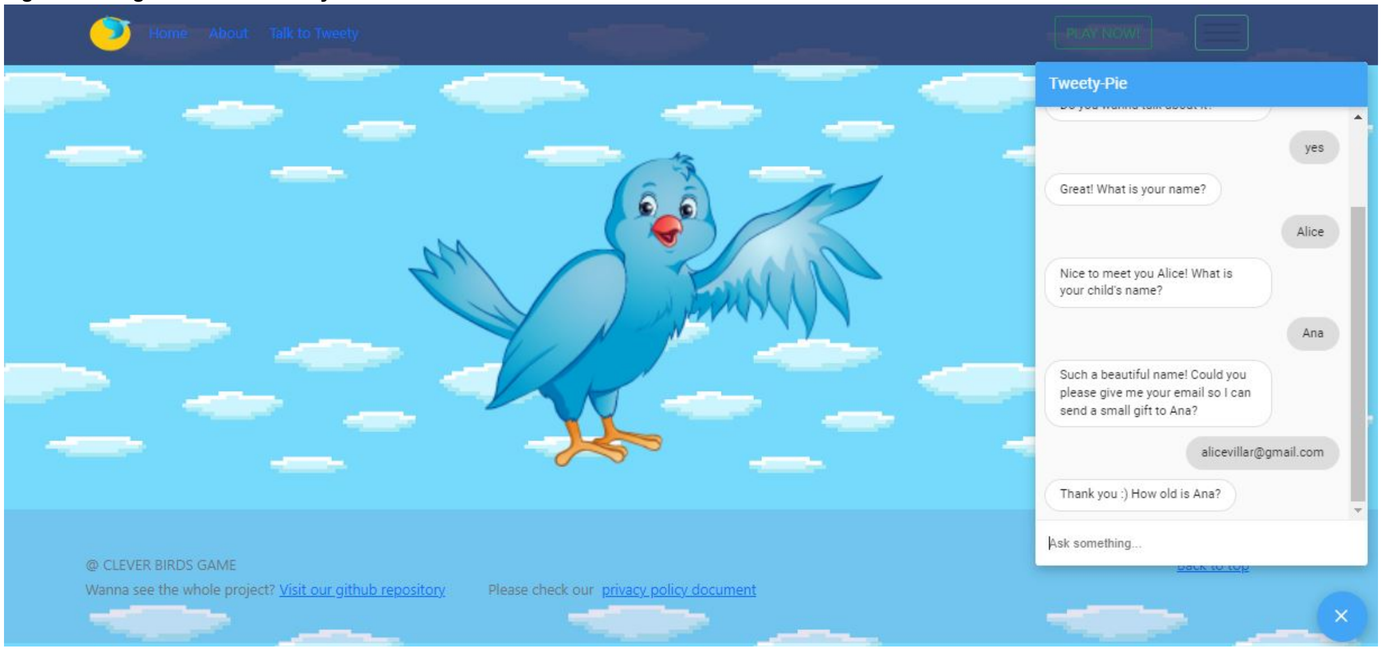
Username
Alice

Score


@ CLEVER BIRDS GAME
Wanna see the whole project? Visit our github repository Please check our privacy policy document
Back to top

Greetings! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wanna talk about it?

Fig. 7: Talking to chatbot Tweety



Tweety-Pie

yes

Great! What is your name?

Alice

Nice to meet you Alice! What is your child's name?

Ana

Such a beautiful name! Could you please give me your email so I can send a small gift to Ana?

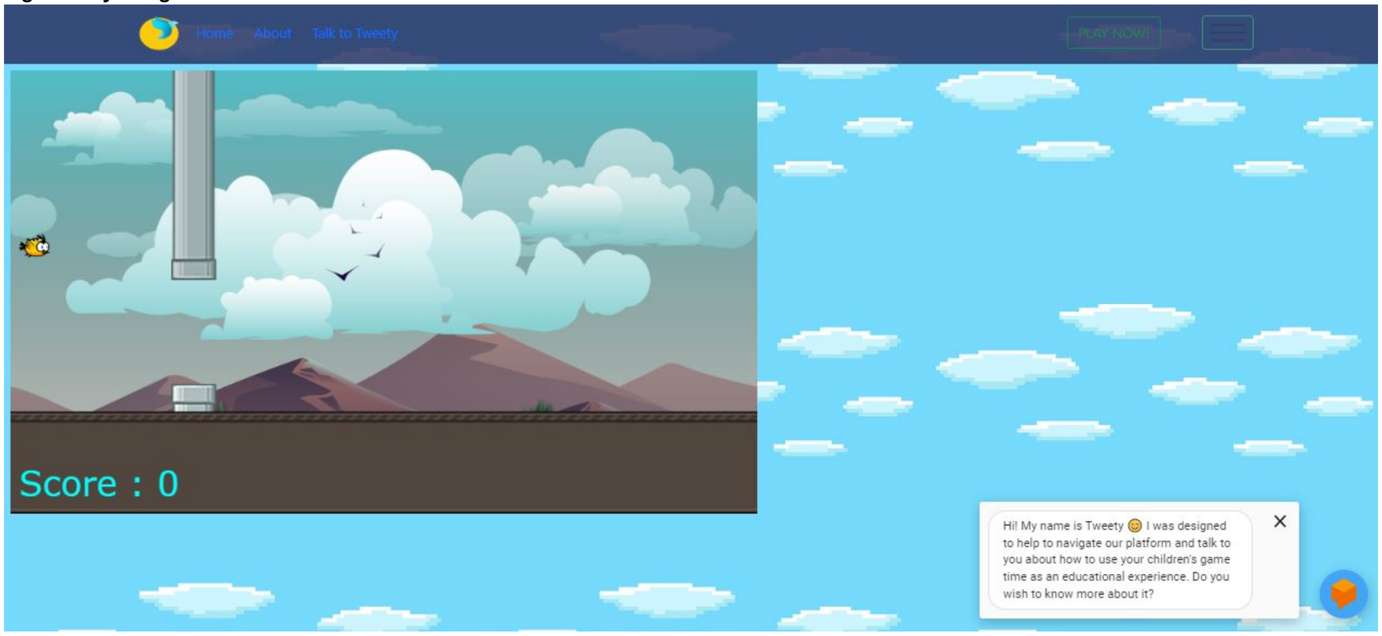
alicevillar@gmail.com

Thank you :) How old is Ana?

Ask something...

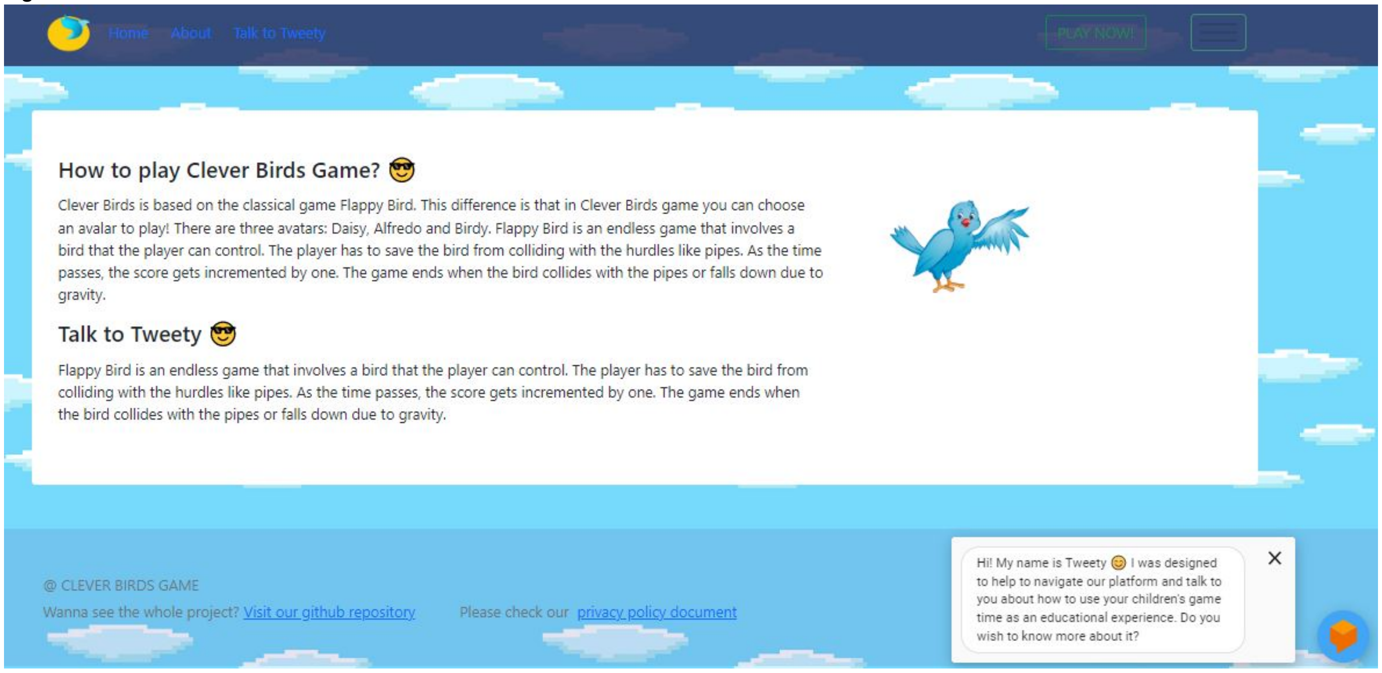
@ CLEVER BIRDS GAME
Wanna see the whole project? Visit our github repository Please check our privacy policy document

Fig. 8: Play the game



Hi! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wish to know more about it?

Fig. 9: About



How to play Clever Birds Game? 😎

Clever Birds is based on the classical game Flappy Bird. This difference is that in Clever Birds game you can choose an avalar to play! There are three avatars: Daisy, Alfredo and Birdy. Flappy Bird is an endless game that involves a bird that the player can control. The player has to save the bird from colliding with the hurdles like pipes. As the time passes, the score gets incremented by one. The game ends when the bird collides with the pipes or falls down due to gravity.

Talk to Tweety 😎

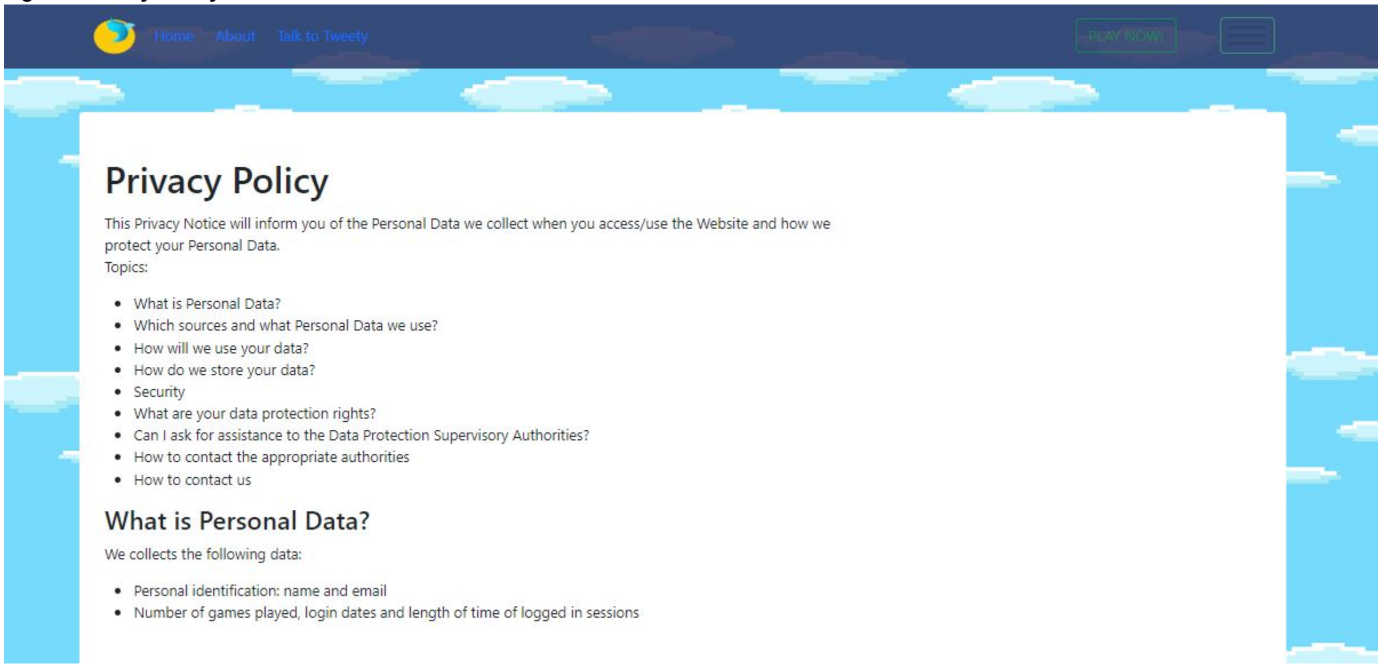
Flappy Bird is an endless game that involves a bird that the player can control. The player has to save the bird from colliding with the hurdles like pipes. As the time passes, the score gets incremented by one. The game ends when the bird collides with the pipes or falls down due to gravity.

© CLEVER BIRDS GAME

Wanna see the whole project? Visit our github repository Please check our privacy policy document

Hi! My name is Tweety 😊 I was designed to help to navigate our platform and talk to you about how to use your children's game time as an educational experience. Do you wish to know more about it?

Fig.10: Privacy Policy



Privacy Policy

This Privacy Notice will inform you of the Personal Data we collect when you access/use the Website and how we protect your Personal Data.
Topics:

- What is Personal Data?
- Which sources and what Personal Data we use?
- How will we use your data?
- How do we store your data?
- Security
- What are your data protection rights?
- Can I ask for assistance to the Data Protection Supervisory Authorities?
- How to contact the appropriate authorities
- How to contact us

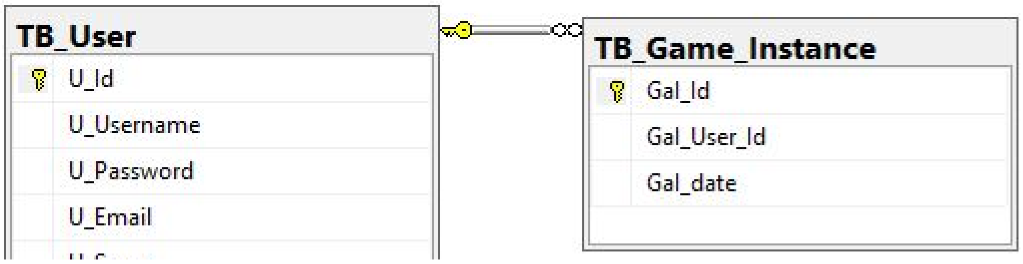
What is Personal Data?

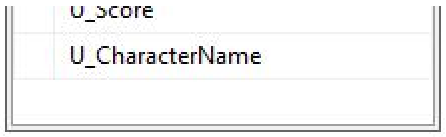
We collects the following data:

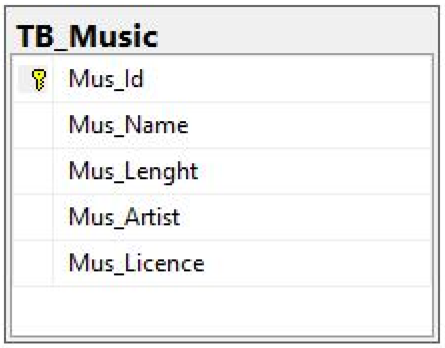
- Personal identification: name and email
- Number of games played, login dates and length of time of logged in sessions

Database Design

Our database design aims to fulfil the definition of the Third Normal Form (Connolly & Beg, 2015). See our entity-relationship diagram, automatically generated with SQL-Server Management Studio:




U_Score
U_CharacterName

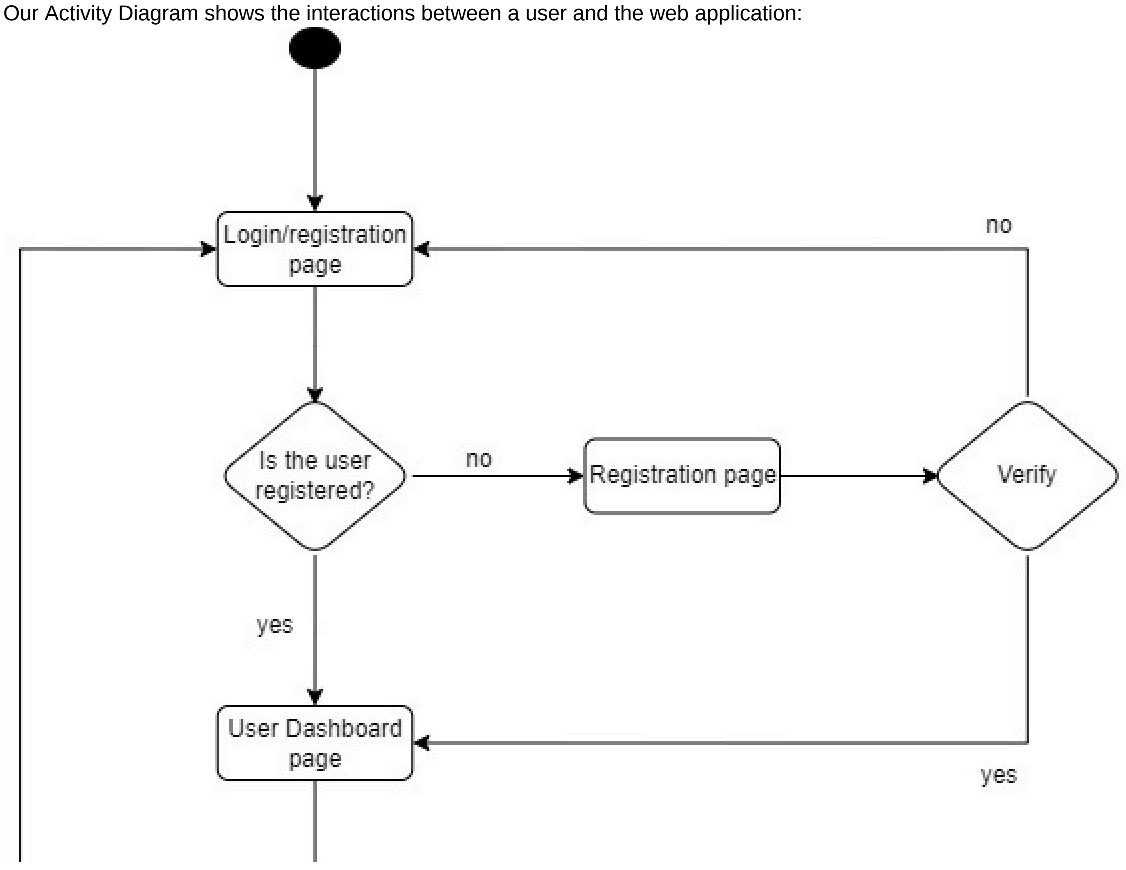

TB_Music
Mus_Id
Mus_Name
Mus_Lenght
Mus_Artist
Mus_Licence

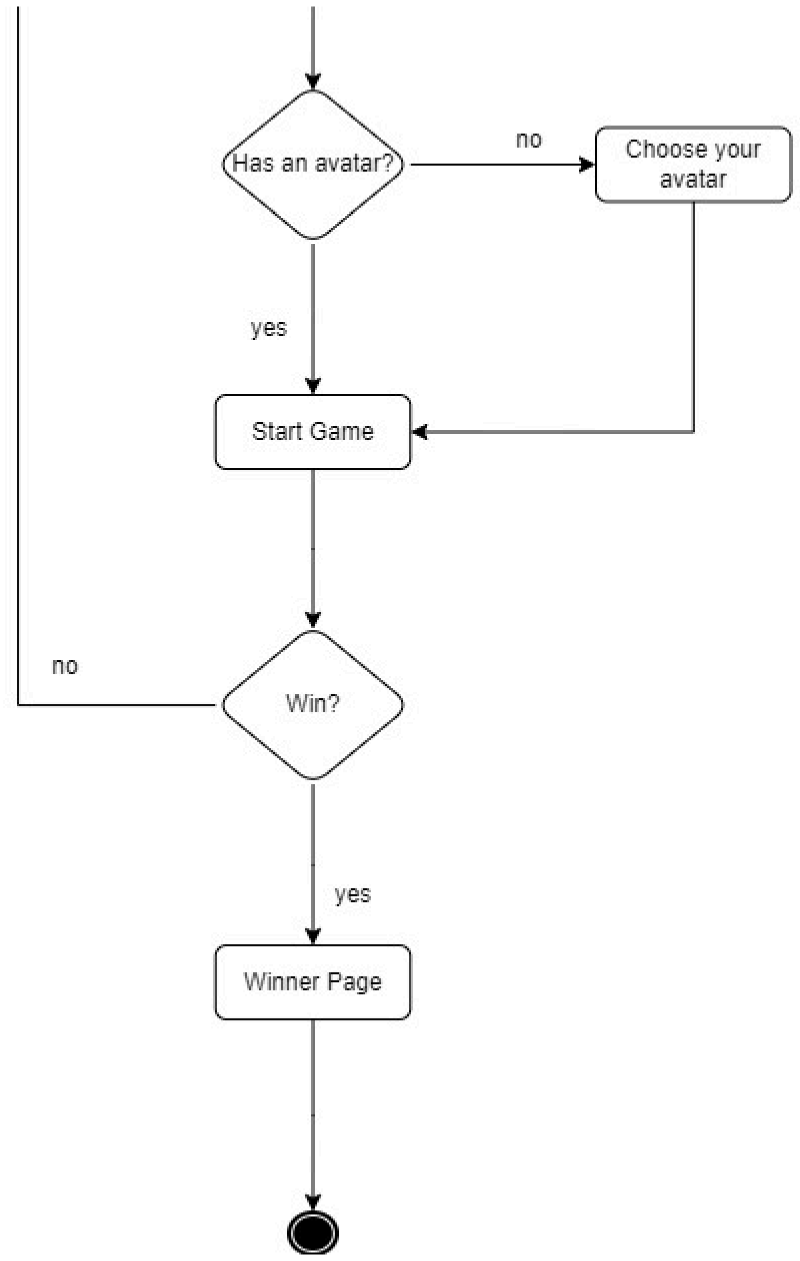
Architectural Design

Clean Code	Best practices, such as: preferring meaningful variable names over writing comments; ensuring all the tests can run.
Programming Paradigm	Flask application using OOP.
Design Pattern	Decorator pattern, for adding responsibilities to objects dynamically.
Design Principles	Apply SOLID principles.
Architectural Styles	Monolithic architecture (the application is combined into a single platform)
Architectural Pattern	Model-View Controller (MVC), dividing the application into three components: model, view, and controller.

Procedural design

Our Activity Diagram shows the interactions between a user and the web application:





5 - Build

Quick Overview

Frameworks	• Flask for the main application • Angular for the game
Testing Frameworks	• pytest for python unit and integration tests • jest for javascript unit and integration tests • Cypress for e2e tests?
Linting	• *Pylint* to implement best practices, identify code violations and provide refactoring suggestions.
Platform	Docker
Repository	https://github.com/SEPM-2022/CleverBirds

Project structure

The project follows the Model-View-Controller (MVC) pattern, an architectural pattern consisting of three parts: Model, View, Controller. Each of them has specific responsibilities. This "separation of concerns" provides for a better division of labor and improved maintenance.

This can be seen through directories in the root directory of the project:

- **models**

Handles data logic.

- **controllers**

Controller part of the application. These get the information from the models and in turn display that information by connecting with the templates (View).

- **templates**

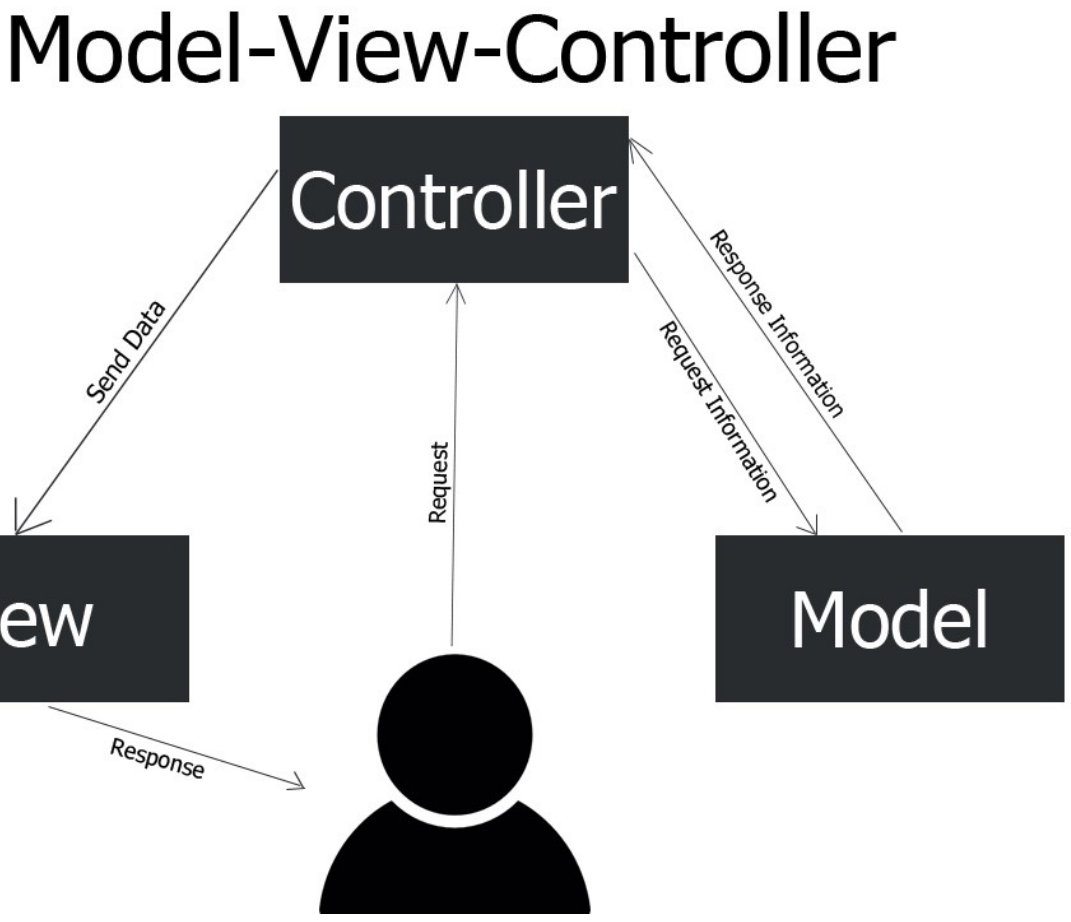
It’s the “view” part of the application. It displays the information from the model to the user.

- **assets**

Stylesheets, images, javascript and mock files.

- **tests**

Contain unit tests.



MVC Overview

Advantages of MVC

- MVC architecture will separate the user interface from business logic and business logic
- Components are reusable.
- Easy o maintain.
- Different components of the application in MVC can be independently deployed and maintained.
- This architecture helpt to test components independently.

REFERENCES:

Svirca, Zanfina (2020). Everything you need to know about MVC architecture. Available from: https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1

Getting Started

Prerequisites

Docker desktop: https://www.docker.com/products/docker-desktop/

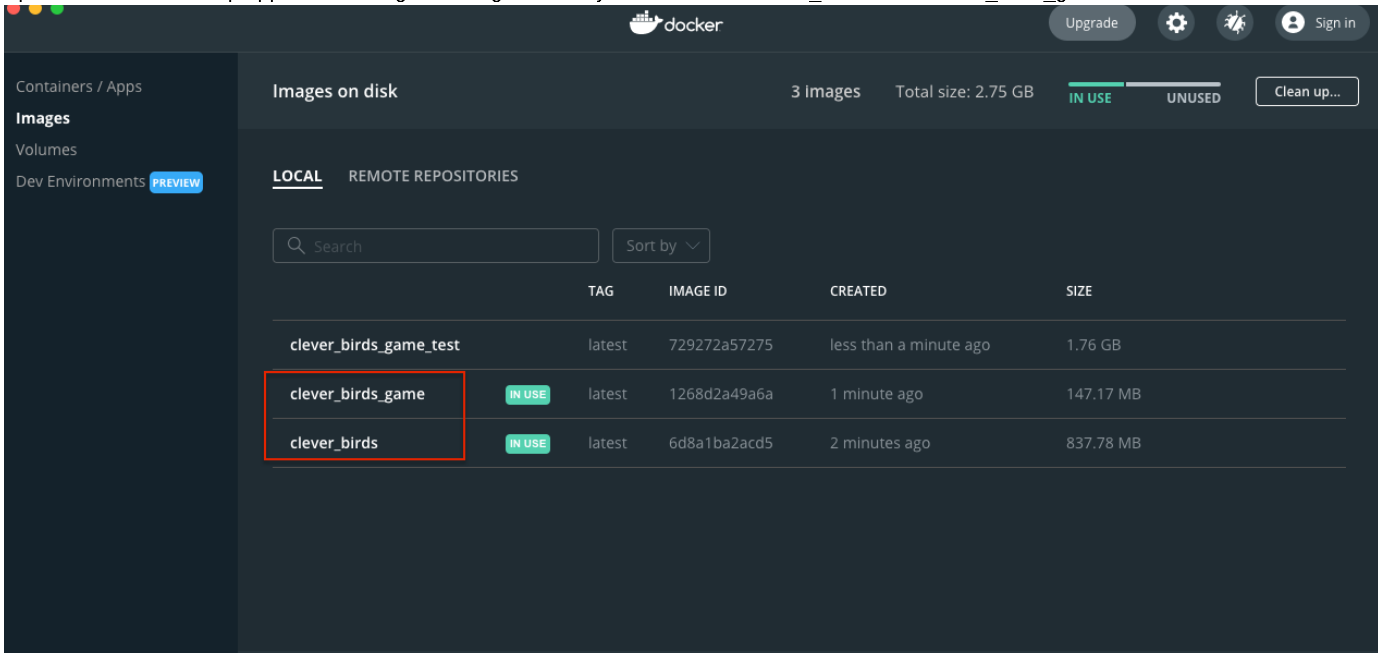
Repository: https://github.com/SEPM-2022/CleverBirds

Run application

After installing docker desktop and cloning the repository, navigate to where the code was copied. In the root of this directory, run the following command:

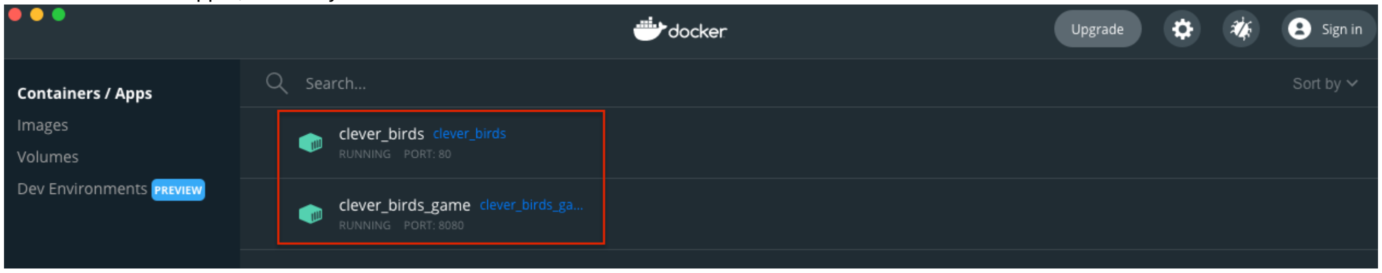
```
sh start.sh
```

Open the docker desktop application and go to “Images“. Here you should see “clever_birds“ and “clever_birds_game“ listed:



		TAG	IMAGE ID	CREATED	SIZE
clever_birds_game_test		latest	729272a57275	less than a minute ago	1.76 GB
clever_birds_game	IN USE	latest	1268d2a49a6a	1 minute ago	147.17 MB
clever_birds	IN USE	latest	6d8a1ba2acd5	2 minutes ago	837.78 MB

Under “Containers/Apps“, the newly created containers should show:



clever_birds clever_birds
RUNNING PORT: 80

clever_birds_game clever_birds_ga...
RUNNING PORT: 8080

The application should now be available at http://localhost.

Ngrok

Ngrok is a cross-platform application that exposes local server ports to the Internet. We will share a ngrock link during our presentation.

- **Download:**

https://ngrok.com/

- **Documentation:**

https://ngrok.com/docs

- **How to create a ngrok link:**

1. $ ./ngrok.exe config add-authtoken 1srifoH59zdF0kFLejqZFvmGLmU_xHHYoS6gLgKatTjvpSdH
2. $ ./ngrok.exe config upgrade
3. $ ./ngrok.exe http 5000

Testing

Unit & Integration testing

In the root directory, the unit and integration tests can be executed with the following command:

```
sh test.sh
```

E2E testing

Before executing the e2e tests, make sure that the application is running on localhost(see Run application section). In the root directory, the e2e tests can be executed with the following command:

```
sh test-e2e.sh
```

Videos of the tests performed are stored under e2e/cypress/videos.

6 - Testing

This page contains our Master Test Plan and our goal is to facilitate the successful planning, controlling, and management for all testing aspects for our project.

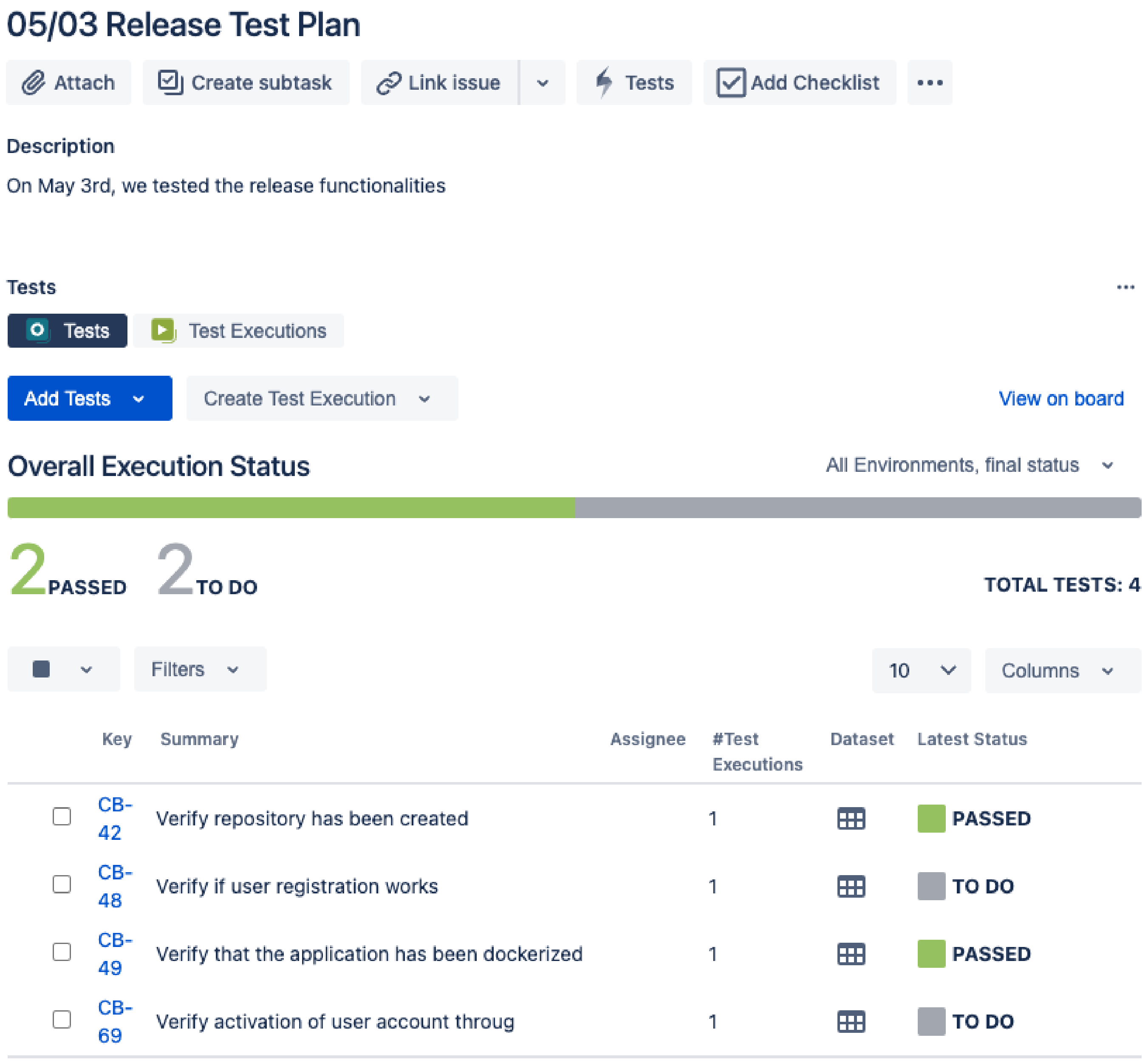
Testing Framework

Scope	• Unit testing, Integration testing, User acceptance testing
Test principles	• Testing processes will be well defined, yet flexible, with the ability to change as needed. • Testing will be a repeatable, quantifiable, and measurable activity.
Test objectives	• Produce documentation that describes how the tester will verify that the system works as intended. • The system is easy to use by the end-users.
Test types	• Manual tests • Automated tests
Defect Management	• The team will report the defects to the project manager with a snapshot JPEG format.

Test documentation	• Github
Tools	• Flask-Testing and Pytest Modules. • Jest testing framework for the game. • Cypress for end-to-end testing. • Jira for bug tracking.

Jira Test Management

We used Jira Test Management to help you test, plan, track, and release our software. Here is a demonstration of our Release test Plan:



05/03 Release Test Plan

Description

On May 3rd, we tested the release functionalities

Tests

Overall Execution Status

2 PASSED 2 TO DO TOTAL TESTS: 4

Key	Summary	Assignee	#Test Executions	Dataset	Latest Status
CB-42	Verify repository has been created		1		PASSED
CB-48	Verify if user registration works		1		TO DO
CB-49	Verify that the application has been dockerized		1		PASSED
CB-69	Verify activation of user account throug		1		TO DO

Feature Testing

From the agreed ten requirements received, we have chosen five as high priority due to their importance to the overall product functionality and design. These are the features to be tested:

a2) User Profile: Users can create a user profile.

a3) Suitable for kids 5-8 years old: Clever Birds is very simple. After signing in, the user can see a green button on the top right ("PLAY NOW!") on all screens. By pressing the button, the user starts playing.

a5) Player persona: On the page “Choose your avatar', the application allows users to choose between three avatars: Daisy, Birdy, and Alfredo.

a6) Cancelation of subscription: On the page "Manage Account", users can delete their account, which will erase their data from our database.

a7) Safety: The chatbot "Tweety" teaches parents how to protect their children by discussing video games' health problems and how to solve them.

Unit & Integration Testing

Instructions

Go to the root directory of the project and run the following command: sh test.sh

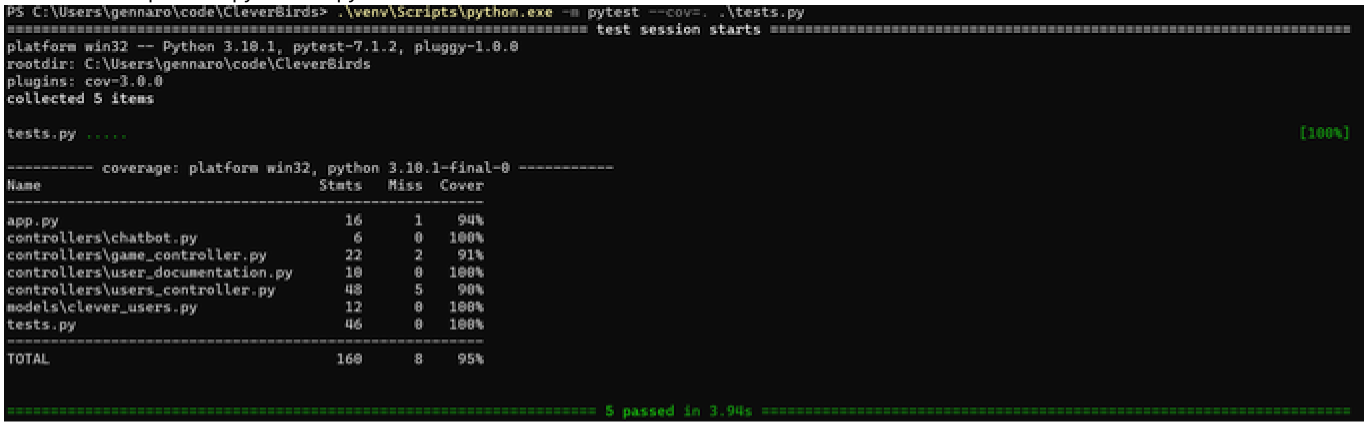
This command will run all unit and integration tests for both the main application as well as the game.

Artifacts

- Here is our pytest report:

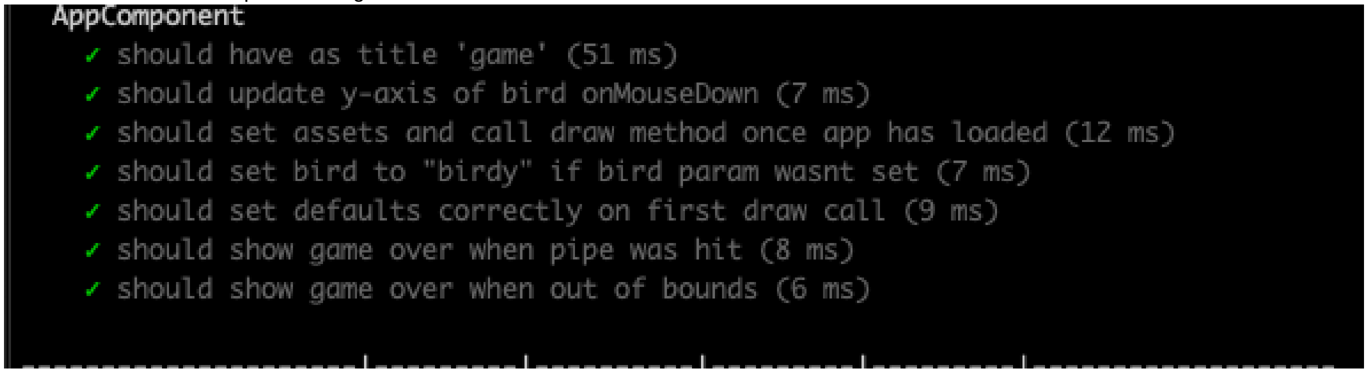
Test Report.pdf

- The output from pytest and pytest-cov module:

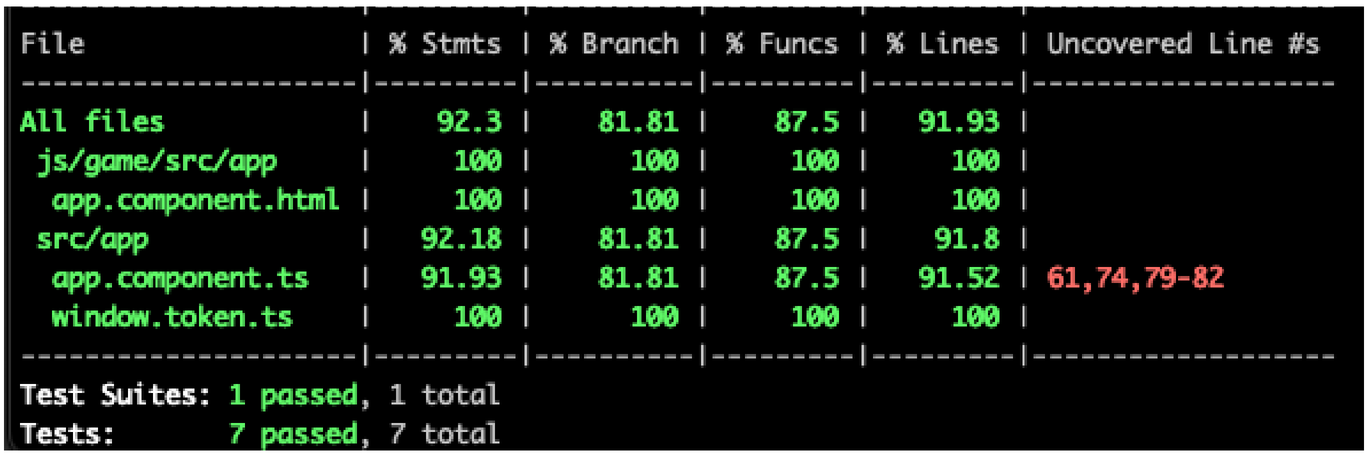


```
PS C:\Users\gennaro\code\CleverBirds> .\venv\Scripts\python.exe -m pytest --cov=. .\tests.py
===================================================================== test session starts =====================================================================
platform win32 -- Python 3.10.1, pytest-7.1.2, pluggy-1.0.0
rootdir: C:\Users\gennaro\code\CleverBirds
plugins: cov-3.0.0
collected 5 items

tests.py .....                                                                                                                                          [100%]

---------- coverage: platform win32, python 3.10.1-final-0 -----------
Name                                    Stmts   Miss  Cover
-----------------------------------------------------------
app.py                                     16      1    94%
controllers\chatbot.py                      6      0   100%
controllers\game_controller.py             22      2    91%
controllers\user_documentation.py          10      0   100%
controllers\users_controller.py            48      5    90%
models\clever_users.py                     12      0   100%
tests.py                                   46      0   100%
-----------------------------------------------------------
TOTAL                                     160      8    95%


====================================================================== 5 passed in 3.94s ======================================================================
```

- Here is the test report for the game:



```
AppComponent
  ✓ should have as title 'game' (51 ms)
  ✓ should update y-axis of bird onMouseDown (7 ms)
  ✓ should set assets and call draw method once app has loaded (12 ms)
  ✓ should set bird to "birdy" if bird param wasnt set (7 ms)
  ✓ should set defaults correctly on first draw call (9 ms)
  ✓ should show game over when pipe was hit (8 ms)
  ✓ should show game over when out of bounds (6 ms)
```



```
---------------------|---------|----------|---------|---------|-------------------
File                 | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
---------------------|---------|----------|---------|---------|-------------------
All files            |    92.3 |    81.81 |    87.5 |   91.93 |
 js/game/src/app     |     100 |      100 |     100 |     100 |
  app.component.html |     100 |      100 |     100 |     100 |
 src/app             |   92.18 |    81.81 |    87.5 |    91.8 |
  app.component.ts   |   91.93 |    81.81 |    87.5 |   91.52 | 61,74,79-82
  window.token.ts    |     100 |      100 |     100 |     100 |
---------------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
```

Getting started with testing

Unit & Integration testing

In the root directory, the unit and integration tests can be executed with the following command:

```
sh test.sh
```

E2E testing

Before executing the e2e tests, make sure that the application is running on localhost(see Run application section). In the root directory, the e2e tests can be executed with the following command:

```
sh test-e2e.sh
```

Videos of the tests performed are stored under e2e/cypress/videos.

User Acceptance Tests

Content:

1. **What?**
2. **Why?**
3. **Who?**
4. **How?**
5. **Acceptance Testing and V-Model**
6. **Best Practices**
7. **Our UAT Plan**

---

1. What?

User Acceptance Testing (UAT) is a type of testing performed by the end-user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration, and system testing is done.

---

2. Why?

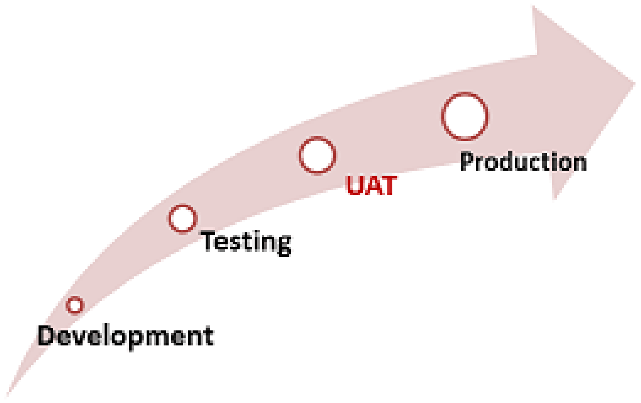
The main purpose of UAT is to validate end-to-end business flow. It does not focus on cosmetic errors, spelling mistakes, or system testing. User Acceptance Testing is carried out in a separate testing environment with a production-like data setup. It is a kind of black-box testing where two or more end-users will be involved.

The need for User Acceptance Testing arises once software has undergone Unit, Integration, and System testing because developers might have built software based on requirements documents by their own understanding, and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

3. Who?

How performs UAT:

- Client
- End users



4. How?

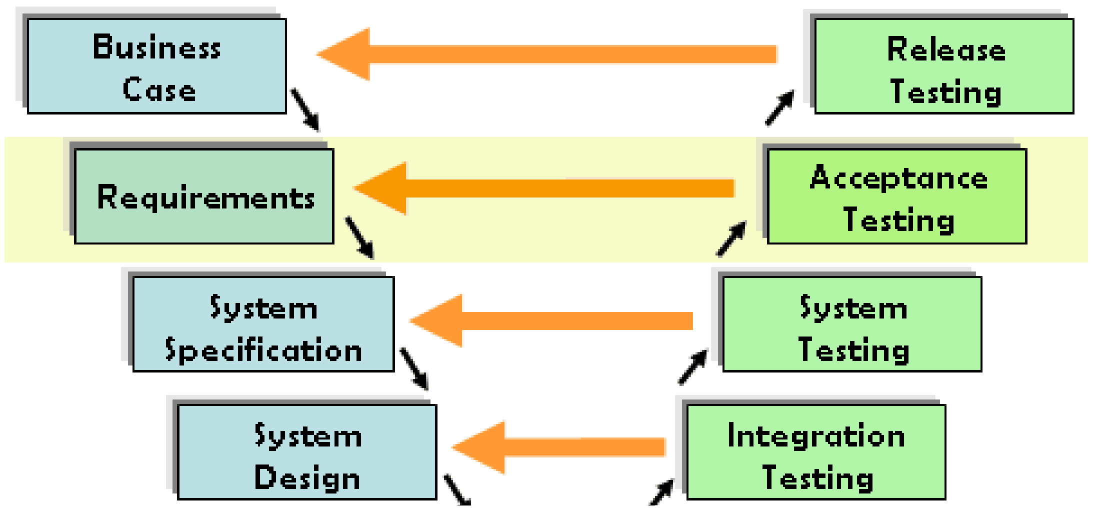
UAT is done by the intended users of the system or software. This type of Software Testing usually happens at the client location which is known as Beta Testing. Once the entry criteria for UAT are satisfied, the following are the tasks that need to be performed by the testers:

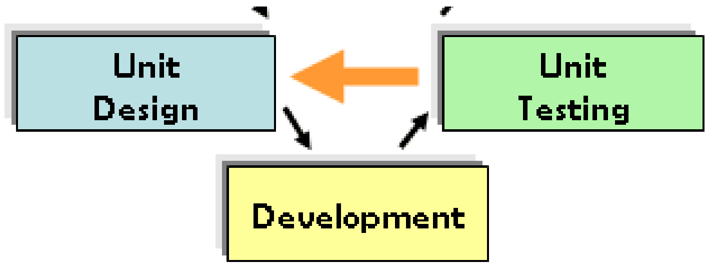
UAT Process

- Analysis of Business Requirements
- Creation of UAT test plan
- Identify Test Scenarios
- Create UAT Test Cases
- Preparation of Test Data(Production like Data)
- Run the Test cases
- Record the Results
- Confirm business objectives

5. Acceptance Testing and V-Model

In VModel, User acceptance testing corresponds to the requirement phase of the Software Development life cycle(SDLC).





6. Best Practices

The following points need to be considered to make UAT Success:

- Prepare UAT plan early in the project life cycle
- Prepare a Checklist before the UAT starts
- Conduct a Pre-UAT session during the System Testing phase itself
- Set the expectation and define the scope of UAT clearly
- Test End to End business flow and avoid system tests
- Test the system or application with real-world scenarios and data
- Think as an Unknown user to the system
- Perform Usability Testing
- Conduct Feedback sessions and meetings before moving to production

7. Our UAT Plan

https://docs.google.com/document/d/1Qh3WDskKqgsYEm1CWnfuVpSQ74RdDzbfhLVtjRiCsrQ/edit

Release Acceptance Testing

Release Acceptance Testing

Release Acceptance Testing is the end stage of the testing process – the final Go/No-Go check performed in a staging environment in advance of a new release.

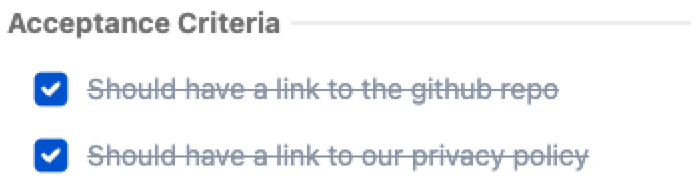
Release testing comprises all the development and testing activities. The test execution lists all the tests that need to pass before a release can be marked as done. Tests are created for every user story, and a story cannot be marked as done if the tests have not passed. To achieve the results mentioned, we will be using a tool called Xray (Xray, N.D.).

Quick overview

Testing tool	Xray (Jira addon)

Process

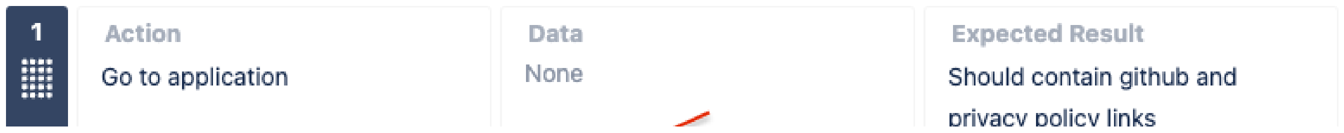
Each user story has acceptance criteria in the form of a list with checkboxes to be ticked if completed by the developer. An example can be seen below:



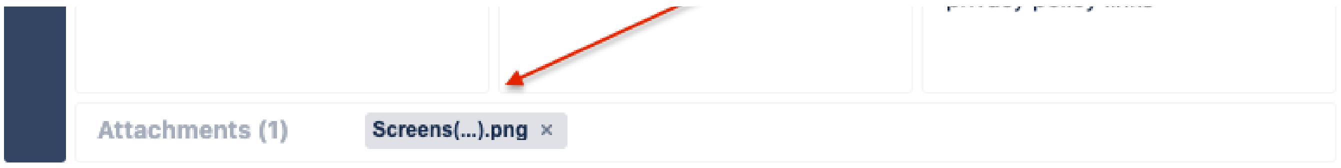
Acceptance Criteria

- ✓ ~~Should have a link to the github repo~~
- ✓ ~~Should have a link to our privacy policy~~

For each user story, in order to tick the acceptance criteria, test cases are created using Xray.
The developer then has to create the tests so that by passing the test, the criteria can be marked as done.



1	Action	Data	Expected Result
	Go to application	None	Should contain github and privacy policy links



Each test should have an uploaded artifact proving the acceptance criteria have been met. Once the tests pass, then the acceptance criteria can be ticked and the user story can be moved to done.

Testing Summary

Blackbox testing:

- Developer/Feature tests (the xray tests in jira),
- E2E tests (you can see how to execute them in the "Getting started" section under "Build" in confluence). Proof of tests are then automatically saved under the e2e/cypress/videos directory for further viewing
- UATs

Whitebox testing:

- Unit & integration testing for the main app
- Unit & integration testing for the game

7 - Releases

Quick overview

No	Release	Start Date	End Date	Description
1	Infrastructure, Login & Registration Release 05 /03	04/04/22	03/05/22	This release cover the infrastructure, login and registration modules to be completed
2	Game & Chatbot Release 05/18	03/05/22	18/05/22	This release covers the game and chatbot modules to be completed

Release Epics

Release No	Epic	Status
1	Development Infrastructure	Done
1	User Registration & Login	Done
2	Game	Done
2	Chatbot	Done

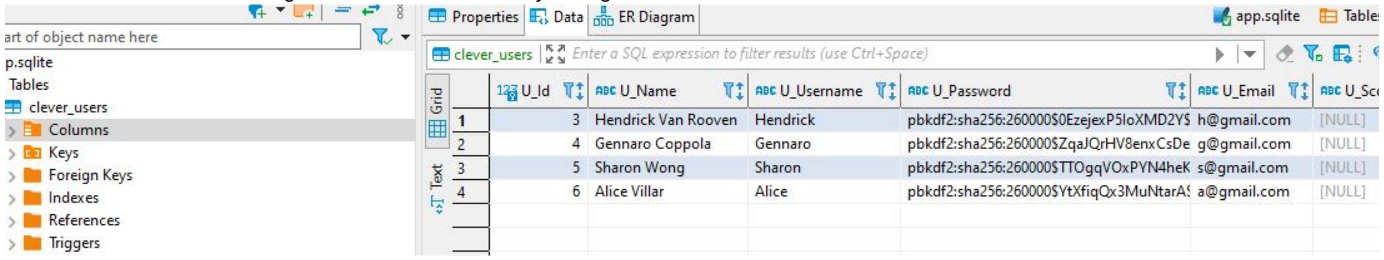
8 - Training & Support

To keep our team sharp and up to date, we created a number of pages for training and support.

DBeaver

DBeaver is a free, open-source multiplatform database management tool and SQL client for developers and database administrators. We will be using this tools as an interface for SQlite. It will be useful to test, document and present our Database performance.

In this PrintScreen I'm showing our database successfully storage of four new users:



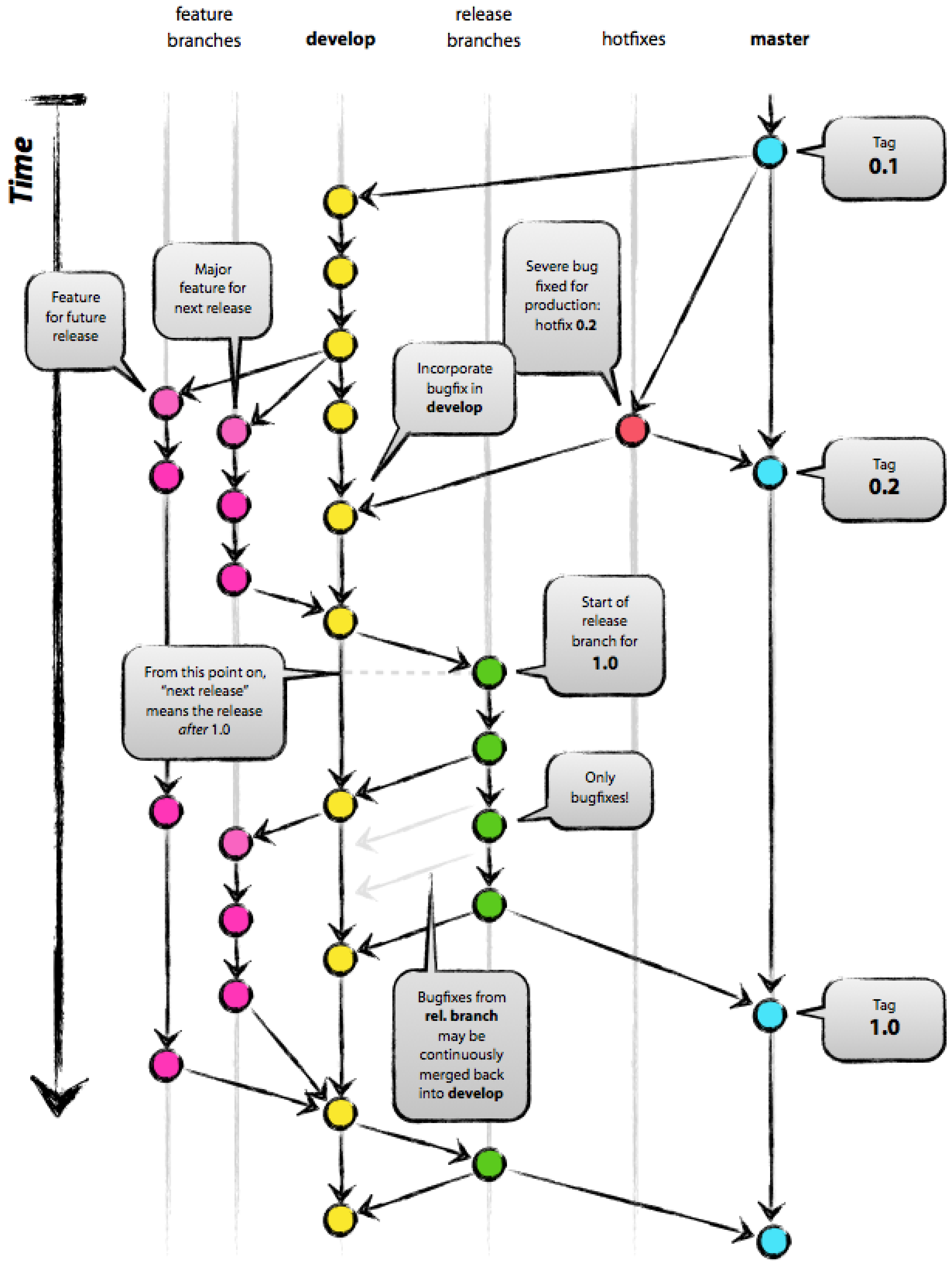
	U_Id	U_Name	U_Username	U_Password	U_Email	U_Sc
1	3	Hendrick Van Rooven	Hendrick	pbkdf2:sha256:260000$0EzejexP5loXMD2Y$	h@gmail.com	[NULL]
2	4	Gennaro Coppola	Gennaro	pbkdf2:sha256:260000$ZqaJQrHV8enxCsDe	g@gmail.com	[NULL]
3	5	Sharon Wong	Sharon	pbkdf2:sha256:260000$TTOgqVOxPYN4heK	s@gmail.com	[NULL]
4	6	Alice Villar	Alice	pbkdf2:sha256:260000$YtXfiqQx3MuNtarA	a@gmail.com	[NULL]

Link to download the tool: Download | DBeaver Community

Git flow bug fix

Here is the tutorial our team followed to do a bug fix:

https://git.logikum.hu/flow/bugfix


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Tag 1.0
Bugfixes from rel. branch may be continuously merged back into develop

REFERENCES:

https://nvie.com/posts/a-successful-git-branching-model/

GitHub Project Guidelines

Here are the guidelines our team decided to follow in order to properly document our project on GItHub:

https://gist.github.com/rsp/057481db4dbd999bb7077f211f53f212

Flask

Here some of the resources we used to build a Python application with Flask:

https://flask.palletsprojects.com/en/2.0.x/quickstart/

https://www.digitalocean.com/community/tutorials/how-to-structure-large-flask-applications

https://www.youtube.com/watch?v=wIy0Gyz2Jlo

https://www.youtube.com/watch?v=3mwFC4SHY-Y

9 - User Documentation

Content:

1. **What?**
2. **How?**
3. **Best Practices**
4. **Our User Documentation**

---

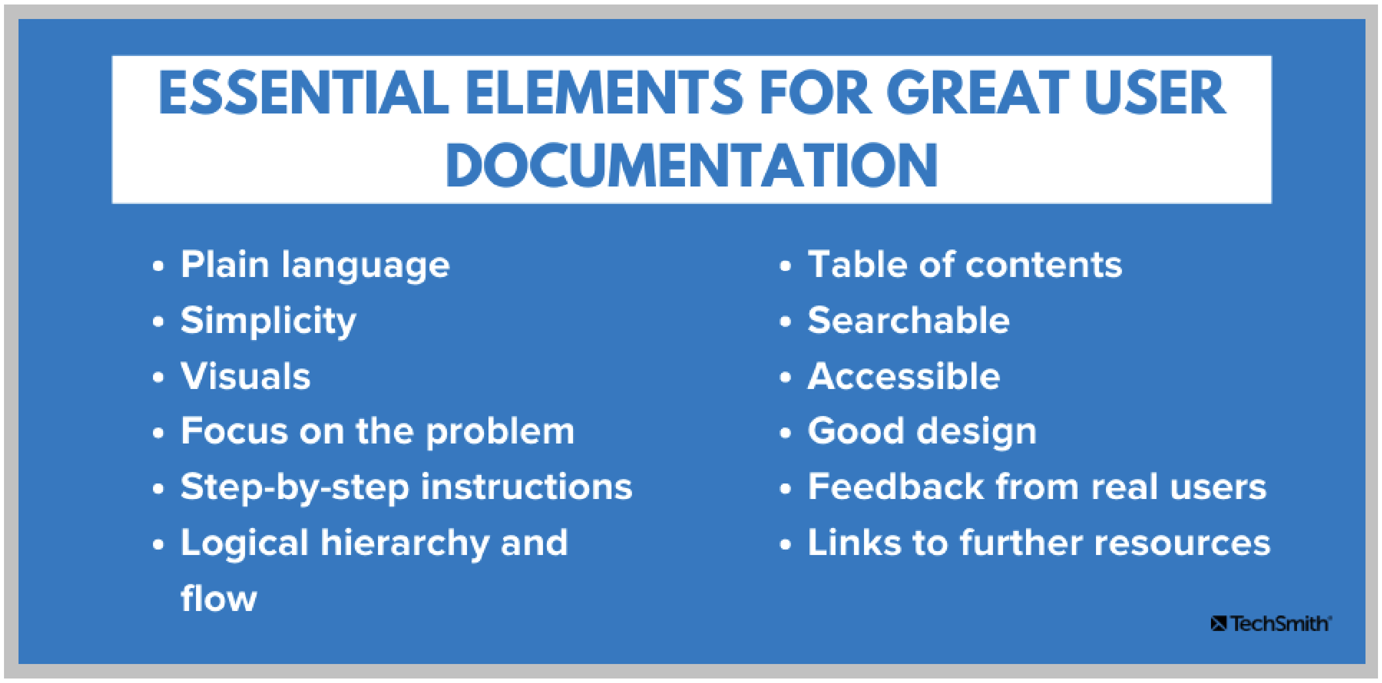
1. What?

User documentation is the content that you provide the end user in order for them to be more successful with your product or service. Also known as user guides, instruction manuals, or user manuals, user documentation is there to hold your customer's hand as they learn about your product.

2. How?

Types of user documentation include training manuals, user manuals, release notes and installation guides

3. Best Practices



ESSENTIAL ELEMENTS FOR GREAT USER DOCUMENTATION

- Plain language
- Simplicity
- Visuals
- Focus on the problem
- Step-by-step instructions
- Logical hierarchy and flow
- Table of contents
- Searchable
- Accessible
- Good design
- Feedback from real users
- Links to further resources

4. Our User Documentation

Our User Documentation includes:

1. Readme file on Github: https://github.com/SEPM-2022/CleverBirds

2. In our application:

- Page “about us”
- Page “privacy policy”