Advanced Software Development – Group Report

GitLab Repository

https://gitlab.uwe.ac.uk/ak2-hassanmahm/horizon-cinemas

Software and Methodology Decisions

Software Development Methodology

Agile development methodologies lend themselves well to short development time frames, small teams and flexibility. Due to the size of our team and the given development deadline, regular meetings combined with quick changes in direction and frequent progress evaluations were crucial to delivering cohesive teamwork and a satisfactory end product. For this reason, we opted for the SCRUM methodology and elected Amro Mahmoud as our Scrum Master. Jira was used as a collaboration and organisational tool for our sprints which we kept deliberately short in order to break the project down into easier chunks for rapid redistribution of larger tasks between team members in the event that changes in direction proved favourable.

Database

We decided to follow a non-relational database management system approach for this project. Had we chosen to use a relational database, MySQL would have been an adequate alternative. Other RDBMS options such as SQLite and PostgreSQL, whilst popular, restrict functionality and growth and are ill fitted to the proposal. SQLite cannot accommodate a client-server and is incompatible with the necessity for multiple user types to have varying levels of access to the system. PostgreSQL hinders development speed by simply being more extensive capability-wise and therefore, is marginally more challenging to learn than MySQL. It also has significant drawbacks regarding its memory performance in relation to how many client connections are required of it. With these considerations in mind, MySQL finds a comparative middle ground.

However, given the specified time frame and particular project requirements we determined that a document-oriented database management system would be most suitable. With extensive resources and a wide user base, MongoDB provided a manageable learning curve whilst ultimately enabling quicker utilisation. Furthermore, the versatility of both vertical and horizontal scaling allows the technology (and clientele) room for rapid future growth. Congruous with the concurrent read/write operations staff will be performing when using the technology and capable of establishing consistency of data via concurrency control measures, MongoDB provides a strong case for being an appropriate software solution for the specified project parameters as well as for immediate and potential ongoing client objectives.

Reflections on the Process

Identifying Assignment Criteria

Our principle goal of identifying, understanding and breaking down the problem underwent constant iterations, even whilst taking less precedence as progress advanced. Time invested in this endeavour during the initial design phase afforded us greater clarity of direction and less potential for unaffordable loss of time due to errors and miscalculation during later phases.

Task Distribution

After the main design phase and analysis of the objectives/challenges and with Amro Mahmoud acting as Scrum Master, initial task allocation began as such:

Farhan Shakeel:

- Database, Backend/Business Logic, Testing.

Vanessa Brown:

- GUI, Backend/Business Logic.

Amro Mahmoud:

- Sprint planning and backlog management
- Database, GUI and Backend/Business Logic (as required).

This is a general overview of our individual contributions. Further details will be provided in our individual reports.

Software and Design Challenges

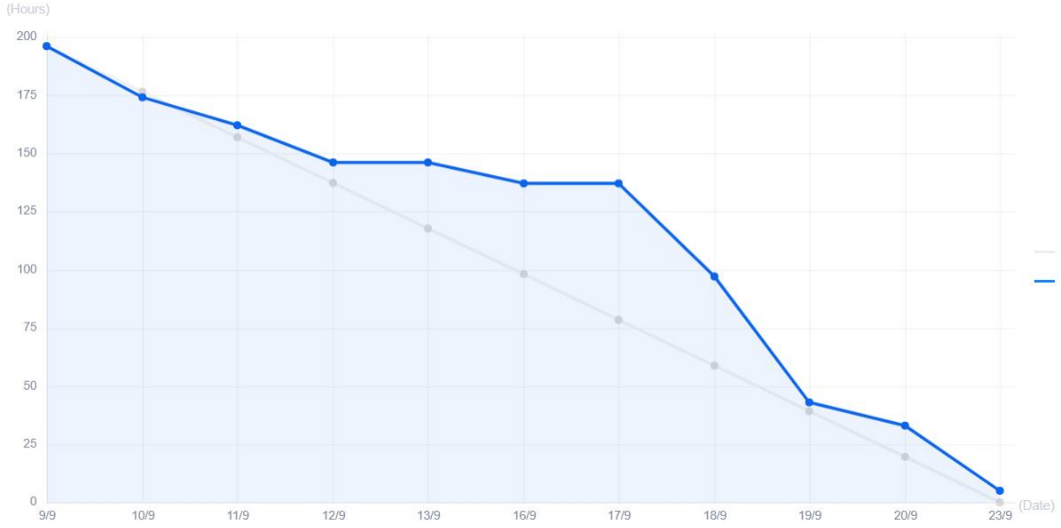
We encountered numerous problems during GUI design, predominantly concerning scaling and layout challenges complicating our attempts at uniformity across various screens and devices. Unfortunately, this encompassed a considerable portion of our development time. In hindsight and if given the option, we would have eschewed the use of Tkinter for the GUI implementation entirely.

Team Discussions

Team meetings were held via the Agile convention. Topics of discussion included: Latest implementations/tasks achieved; subsequent and future implementations/tasks to be done, problems/challenges encountered and methods for solving them; sprint allocation and general reflections on our progress/the process.

Task distribution and planning presented challenges with respect to viable current tasks verses subsequent tasks. Many implementations were dependant on one another and some wasted effort/time resulted from the combined effect of task management and decreased/interrupted communications during the Christmas break. Originally, our meetings were held on a weekly basis with more frequent intermediary communications happening via message. Internet connection issues and illness effectively reduced the ease with which we could update one another, plan ahead and complete work. Despite this, our progress did accelerate when we re-established communications and returned from the break. Our Burndown Chart suffered something much like the example team's Burndown Chart in the below image:

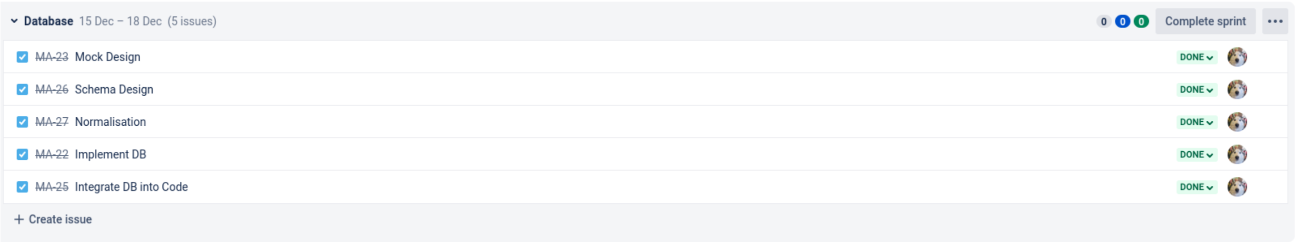
Figure 1 - Example Burndown Chart



ZenTao. (2020)

This was a good learning experience and testament to how important appropriate planning and team organisation is.

Backlog



Database 15 Dec – 18 Dec (5 issues)

0 0 0 Complete sprint ⋯

☑ ~~MA-23~~ Mock Design DONE

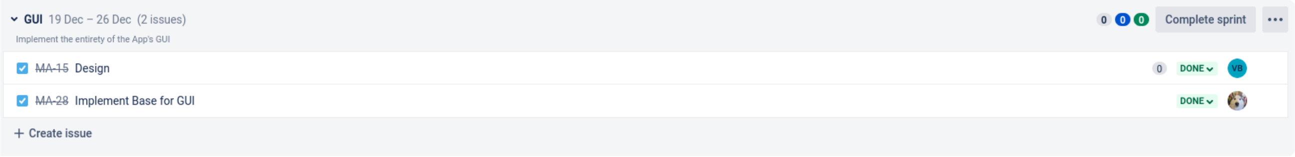
☑ ~~MA-26~~ Schema Design DONE

☑ ~~MA-27~~ Normalisation DONE

☑ ~~MA-22~~ Implement DB DONE

☑ ~~MA-25~~ Integrate DB into Code DONE

+ Create issue



GUI 19 Dec – 26 Dec (2 issues)

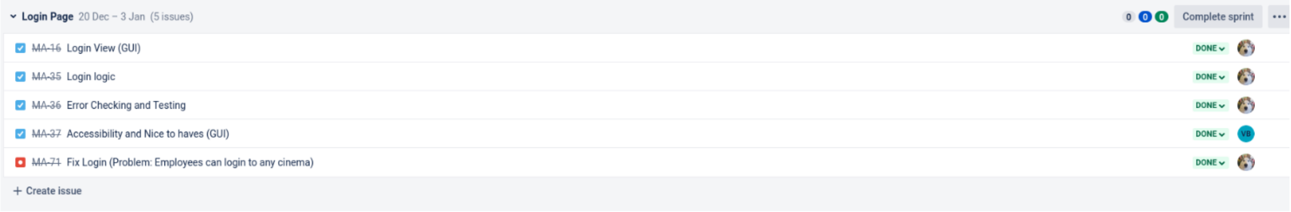
Implement the entirety of the App's GUI

0 0 0 Complete sprint ⋯

☑ ~~MA-15~~ Design 0 DONE VB

☑ ~~MA-28~~ Implement Base for GUI DONE

+ Create issue



Login Page 20 Dec – 3 Jan (5 issues) 0 0 0 Complete sprint ···

☑ ~~MA-16~~ Login View (GUI) DONE

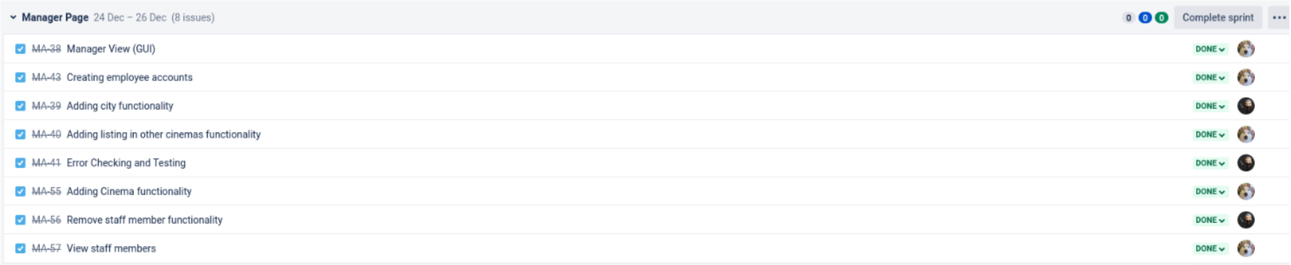
☑ ~~MA-35~~ Login logic DONE

☑ ~~MA-36~~ Error Checking and Testing DONE

☑ ~~MA-37~~ Accessibility and Nice to haves (GUI) DONE VB

🟥 ~~MA-71~~ Fix Login (Problem: Employees can login to any cinema) DONE

+ Create issue



Manager Page 24 Dec – 26 Dec (8 issues) 0 0 0 Complete sprint ···

☑ ~~MA-38~~ Manager View (GUI) DONE

☑ ~~MA-43~~ Creating employee accounts DONE

☑ ~~MA-39~~ Adding city functionality DONE

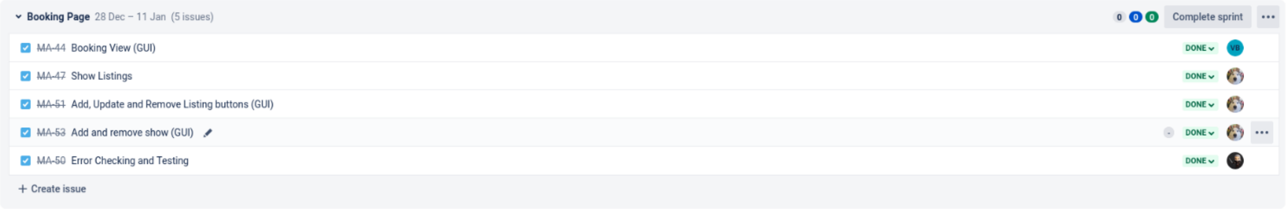
☑ ~~MA-40~~ Adding listing in other cinemas functionality DONE

☑ ~~MA-41~~ Error Checking and Testing DONE

☑ ~~MA-55~~ Adding Cinema functionality DONE

☑ ~~MA-56~~ Remove staff member functionality DONE

☑ ~~MA-57~~ View staff members DONE



Booking Page 28 Dec – 11 Jan (5 issues) 0 0 0 Complete sprint ···

☑ ~~MA-44~~ Booking View (GUI) DONE VB

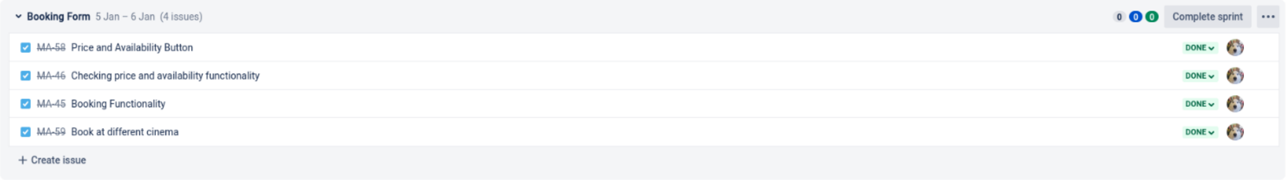
☑ ~~MA-47~~ Show Listings DONE

☑ ~~MA-51~~ Add, Update and Remove Listing buttons (GUI) DONE

☑ ~~MA-53~~ Add and remove show (GUI) DONE ···

☑ ~~MA-50~~ Error Checking and Testing DONE

+ Create issue



Booking Form 5 Jan – 6 Jan (4 issues) 0 0 0 Complete sprint ···

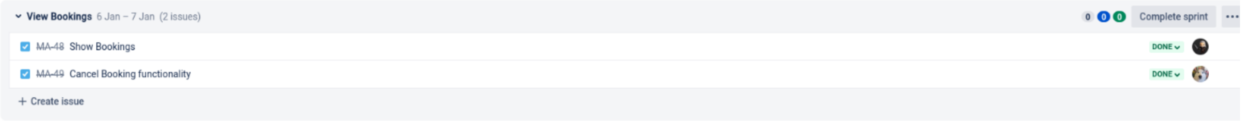
☑ ~~MA-58~~ Price and Availability Button DONE

☑ ~~MA-46~~ Checking price and availability functionality DONE

☑ ~~MA-45~~ Booking Functionality DONE

☑ ~~MA-59~~ Book at different cinema DONE

+ Create issue

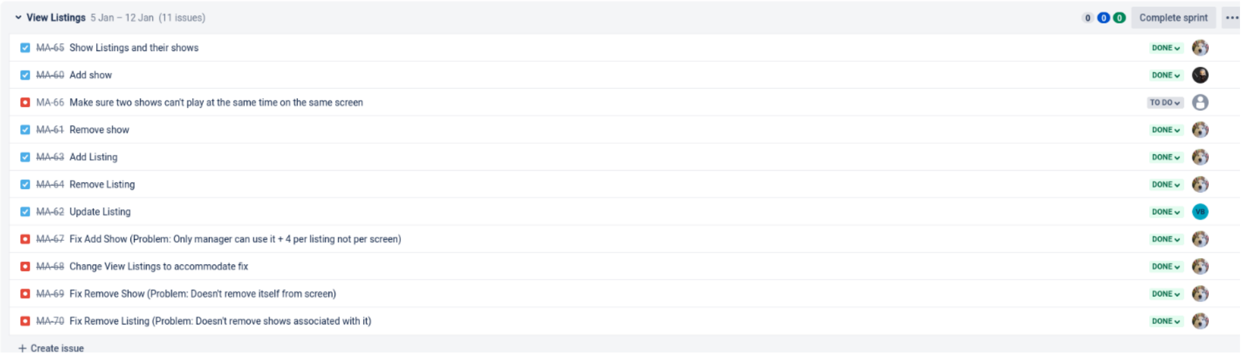


View Bookings 6 Jan – 7 Jan (2 issues) 0 0 0 Complete sprint ···

☑ ~~MA-48~~ Show Bookings DONE

☑ ~~MA-49~~ Cancel Booking functionality DONE

+ Create issue



View Listings 5 Jan – 12 Jan (11 issues) 0 0 0 Complete sprint ···

☑ ~~MA-65~~ Show Listings and their shows DONE

☑ ~~MA-60~~ Add show DONE

🟥 MA-66 Make sure two shows can't play at the same time on the same screen TO DO

☑ ~~MA-61~~ Remove show DONE

☑ ~~MA-63~~ Add Listing DONE

☑ ~~MA-64~~ Remove Listing DONE

☑ ~~MA-62~~ Update Listing DONE VB

🟥 ~~MA-67~~ Fix Add Show (Problem: Only manager can use it + 4 per listing not per screen) DONE

🟥 ~~MA-68~~ Change View Listings to accommodate fix DONE

🟥 ~~MA-69~~ Fix Remove Show (Problem: Doesn't remove itself from screen) DONE

🟥 ~~MA-70~~ Fix Remove Listing (Problem: Doesn't remove shows associated with it) DONE

+ Create issue

References

Axosoft (2012) *Intro to scrum in under 10 minutes*February 20, 2012 [online]. Youtube. Available from: https://www.youtube.com/watch?v=XU0llRltyFM [Accessed 6 January 2023].

“Burndown Chart and what it tells - agile - ZenTao” (no date) *Zentao.pm*. [online]. Available from: https://www.zentao.pm/blog/Burndown-Chart-and-What-It-Tells-781.html [Accessed 9 January 2023].

“Database modeling with UML” (no date) *Sparxsystems.com*. [online]. Available from: https://sparxsystems.com/resources/tutorials/uml/datamodel.html [Accessed 6 January 2023].

Drake, M. (2014a) *A comparison of NoSQL database management systems and models Digitalocean.com*.February 21, 2014 [online]. DigitalOcean. Available from: https://www.digitalocean.com/community/tutorials/a-comparison-of-nosql-database-management-systems-and-models [Accessed 6 January 2023].

Drake, M. (2014b) *SQLite vs MySQL vs PostgreSQL: A comparison of relational database management systems Digitalocean.com*.February 21, 2014 [online]. DigitalOcean. Available from: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems [Accessed 6 January 2023].

Drake, M. (2019) *Understanding database sharding Digitalocean.com*.February 7, 2019 [online]. DigitalOcean. Available from: https://www.digitalocean.com/community/tutorials/understanding-database-sharding [Accessed 6 January 2023].

e-Zest Solutions Inc (no date) *Agile Software Development E-zest.com*. [online]. Available from: https://www.e-zest.com/agile_software_development [Accessed 6 January 2023].

“What are the Different Types of Agile Methodologies?” (no date) *Wrike.com*. [online]. Available from: https://www.wrike.com/project-management-guide/faq/what-are-the-different-types-of-agile-methodologies/ [Accessed 6 January 2023].

“What is an object-oriented database?” (no date) *MongoDB*. [online]. Available from: https://www.mongodb.com/databases/what-is-an-object-oriented-database [Accessed 6 January 2023].