**Document 1: Definition of done (DOD)**

The Definition of Done (DOD) is a set of criteria or conditions that a product increment must meet before it can be considered complete and potentially shippable in Agile and Scrum projects. It serves as a quality standard and ensures that the development team and stakeholders have a shared understanding of what constitutes a finished product increment.

A checklist agreed upon by the team that defines when backlog item is considered complete.

DoD Checklist:

|  |  |  |
| --- | --- | --- |
|  | Code for all functionality implemented | The development team writes all necessary code to implement feature. |
|  | All assumptions in the user story are met. | All pre-defined user story assumptions are considered and fulfilled. |
|  | Project builds successfully without errors. | The code-base must successfully compile/build without any technical errors or missing dependencies. |
|  | Unit test written and passing. | Developers must write unit tests for new features and ensure they pass. |
|  | Feature deployed to test environment. | Code is deployed to a staging environment that mirrors production to allow early testing. |
|  | Cross-browser/ device tests passes. | Features must work on all specified devices and browsers listed in the requirements. |
|  | UX design approval obtained | The design team must review and sign off on the feature to ensure UI/UX consistency. |
|  | QA testing completed; defects resolved | QA engineers test the feature; any defects found must be resolved before marking “done.” |
|  | Acceptance criteria met. | All defined acceptance criteria for the story are satisfied. |
|  | Product owner approval obtained. | The PO validates the story’s functionality and gives the final sign-off. |
|  | Refactoring completed. | Any necessary code clean-up or optimization must be done post-development. |
|  | Change requests are documented. | Any configuration or environment-specific changes are documented for future reference. |
|  | Related documentation updated. | User or system documentation must be updated to reflect new/changed functionality. |
|  | Peer code review conducted. | Code must be reviewed by at least one peer for best practices and code quality. |

**Document 2: Product Vision**

A **Product Vision Document** is a high-level, strategic document that outlines the long-term mission and purpose of a product. It provides direction and clarity for everyone involved in building, marketing, and supporting the product. Think of it as a **north star** that keeps all stakeholders aligned and focused.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Scrum project name: | Hudle application enhancement | | | | | |
| Venue: | Online collaboration via Jira/Confluence + occasional in-person stand-ups (Gurgaon Office) | | | | | |
| Date: | Start time: | | | End time: | | Duration: |
| Client: | Hudle Sports Tech Pvt. Ltd. | | | | | |
| Stakeholder list: | Product Manager, Venue Partners, End Users (Athletes, Sports Groups), Operations & Support Teams | | | | | |
| Scrum team | | | | | | |
| Scrum master: | | Karan | |  | |  |
| Product owner: | |  | |  | |  |
| Scrum developer 1: | |  | |  | |  |
| Scrum developer 2: | |  | |  | |  |
| Scrum developer 3: | |  | |  | |  |
| Scrum developer 4: | |  | |  | |  |
| Scrum developer 5: | |  | |  | |  |
| Vision statement: To create an intuitive and high-performance sports venue booking platform that empowers users across India to discover, book, and manage sports venues and sessions efficiently, anytime, anywhere. | | | | | | |
| Target group | | | Needs | | Product | Value |
| **Market Segment:** Sports tech industry (B2C and B2B2C)  **Target Users:** Individuals booking for personal recreation  Amateur sports teams or corporate groups  Venue partners managing their slots  Event organizers or trainers | | | Eliminate manual processes and reduce booking errors  Provide real-time availability and instant confirmations  Enable group bookings with invite features  Improve payment integration and refund workflows | | A web and mobile-based application that:  Displays venue details, slots, ratings, and amenities  Allows users to search, filter, and instantly book venues  Supports payments, refunds, and wallet options  Sends reminders and updates on booked sessions | Boosts user engagement and retention  Expands partner base by streamlining backend operations  Opens new revenue streams (ads, premium listings)  Aligns with Hudle's mission to organize sports at scale |

**Document 3: User stories**

|  |  |  |  |
| --- | --- | --- | --- |
| User story 1 | Tasks – 1 | | Priority: high |
| VALUE STATEMENT: AS A USER  I WANT TO SEE REAL TIME AVAILABLE SLOTS  SO THAT I CAN BOOK WITHOUT CLASHES | | | |
| BV: 500 | | CP : 8 | |
| Acceptance criteria:   * Br-all mandatory * The system should display only the slots that are available in real-time. * Users should not be able to select or book already reserved slots. * The slots availability should update automatically to prevent double booking.   Alternative flow:   * User can apply filters like: Venue, Date, Sports   Exceptional flow:   * System notifies user that slot is no longer available. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 2 | Tasks – 2 | | Priority: high |
| VALUE STATEMENT: AS A USER  I WANT TO FILTER VENUES  SO THAT I CAN FIND RELEVANT ONE QUICKLY | | | |
| BV: 200 | | CP : 8 | |
| Acceptance criteria:   * Br-all mandatory * The system should allow filtering venues based on location, sport, date, price range and amenities. * Filtered results should update instantly. * User should be able to modify or clear filters anytime   Alternative flow:   * User modifies or reset filters.   Exceptional flow:   * No venues match selected filters. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 3 | Tasks – 3 | | Priority: High |
| VALUE STATEMENT: AS A VENUE PARTNER  I WANT A BETTER REPORTING TOOL  SO THAT I CAN TRACK VENUE PERFORMANCE, USER ACTIVITY AND REVENUE INSIGHTS EFFICIENTLY | | | |
| BV: 200 | | CP : 8 | |
| Acceptance criteria:   * Br-all mandatory * The tool should display metrics like booking per venue, peak hours, revenue trends and user engagement. * Reports should be filtered by date range, venue, sport and time slot. * Reports should be exportable (PDF/Excel) * Visual dashboards (graphs/charts) should be easy to interpret.   Alternative flow:   * User schedules automated reports.   Exceptional flow:   * Data not available for selected parameters. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| User story 4 | Tasks – 4 | | Priority: Medium |
| VALUE STATEMENT: AS A PRODUCT MANAGER AT HUDLE  I WANT A USER ENGAGEMENT TOOL  SO THAT I CAN KEEP USERS INFORMED, MOTIVATED AND COMING BACK TO BOOK MORE SPORTS SESSION | | | |
| BV: 100 | | CP : 5 | |
| Acceptance criteria:   * Br-all mandatory * The tool should allow targeted push notifications and in-app messages. * User should receive reminders for upcoming bookings and follow-ups after missed sessions. * Admins should be able to create campaigns based on user activity, preferences or inactivity.   Alternative flow:   * Automated engagement based on triggers.   Exceptional flow:   * User unsubscribed from notification. | | | |

**Document 4: Agile product owner experience**

As the agile product owner for Hudle, a sports-tech platform managing venue bookings across India, I led the enhancement of the existing application by identifying operational gaps and prioritizing value-driven features. Collaborated closely with cross-functional teams to align product improvements with user needs and business goals.

Key responsibilities:

* Defined and refined the **product backlog** based on insights from venue managers, users, and internal stakeholders.
* Created detailed **user stories**, acceptance criteria, and managed **agile ceremonies** including sprint planning, backlog grooming, and reviews.
* Prioritized features such as **real-time slot availability, venue filtering**, and a **comprehensive reporting dashboard** for operations teams.
* Coordinated with UI/UX designers to ensure **intuitive user flows** and seamless experiences across web and mobile platforms.
* Ensured **incremental delivery** of enhancements through bi-weekly sprints, tracked progress via **JIRA,** and maintained constant feedback loops.
* Partnered with QA teams to validate stories against business logic and usability standards.

Key achievements:

* Delivered a **real-time booking system** that reduced slot clash complaints by 70%.
* Launched an **enhanced filtering module** that improved venue discovery time by 60%.
* Rolled out a **custom reporting tool** leading to data-driven decisions across 50+ managed venues.

**Document 5: Product and sprint backlog and product and sprint burndown chart**

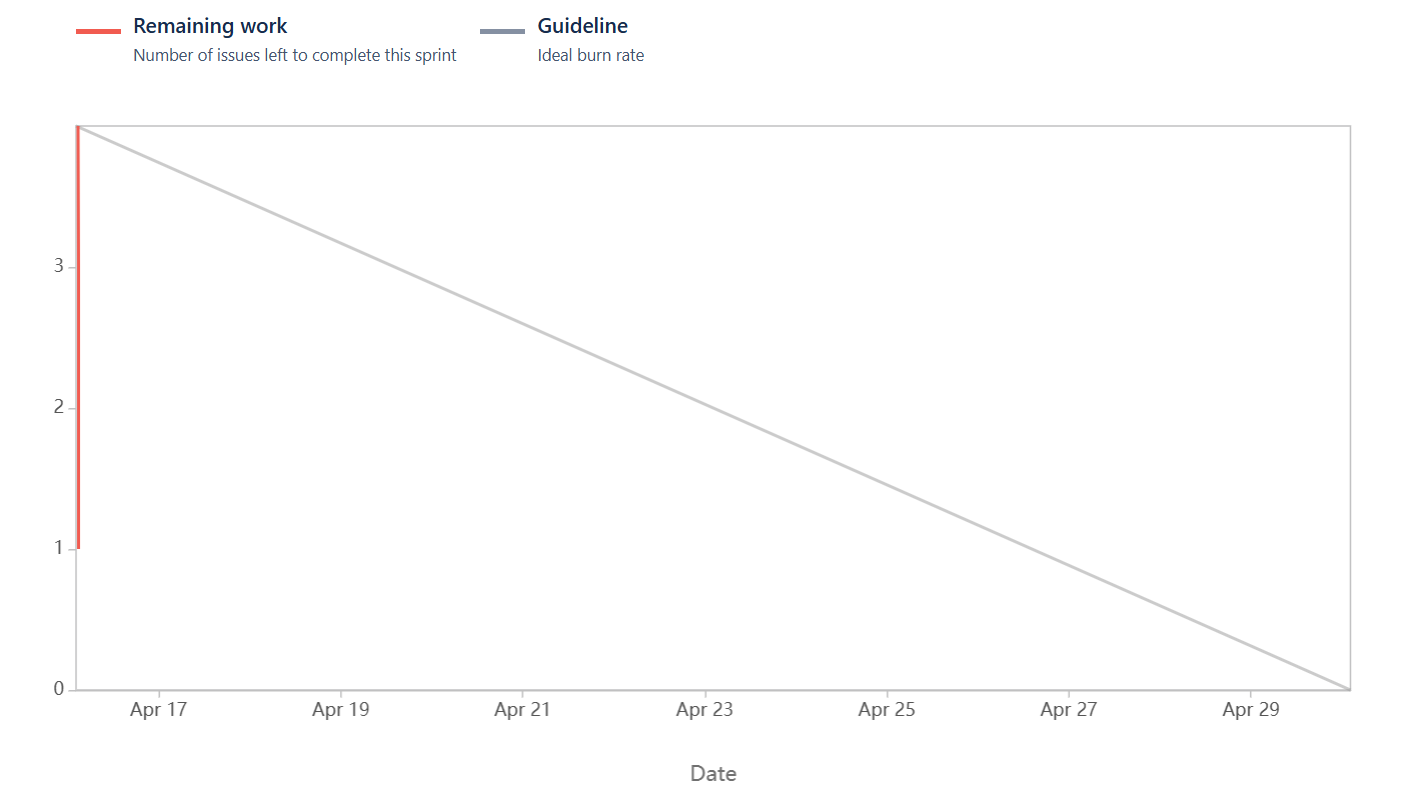
Product backlog

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User story ID | User story | Tasks | Priority | BV | CP | Sprint |
| US001 | Real time availability | API integration, UI development | High | 500 | 8 | Sprint1 |
| US002 | Filter venues | Frontend and backend filter logic | High | 200 | 8 | Sprint1 |
| US003 | Reporting tool | Integrating reporting tool | High | 200 | 8 | Sprint1 |
| US004 | User engagement tool | Engagement tool integration | Medium | 100 | 3 | Sprint1 |

Sprint backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User story ID | User story | Tasks | Owner | Status | Estimated effort |
| US001 | Real time availability | API integration, UI development | Dev1 | Completed | 5 |
| US002 | Filter venues | Frontend and backend filter logic | Dev 2 | Completed | 3 |
| US003 | Reporting tool | Integrating reporting tool | Dev 3 | To do | 5 |

Sprint burndown chart



Product burndown chart

**Document 6: Sprint meetings**

Meeting type 1: Sprint planning meetings

|  |  |
| --- | --- |
| Date | 17/4/2025 |
| Time | 10:00 AM- 11:30 AM |
| Prepared by | PO |
| Attendees | Scrum master, Developers and QA |
| agenda | Finalize sprint goal, assign task and confirm velocity. |

|  |  |  |
| --- | --- | --- |
| Topic | Presenter | Time |
| Sprint goal | PO | 15 minutes |
| Story review | Developer | 30 minutes |
| Task breakdown | Developer | 30 minutes |

|  |  |
| --- | --- |
| Other information: | Focus on user engagement features and reporting tools |
| Observers: | None |
| Resources: | Jira, velocity chart |
| Special notes: | Sprint duration is 2 weeks |

Meeting type 2: Sprint review meeting

|  |  |
| --- | --- |
| Date | 29/4/2025 |
| Time | 11:00 AM- 12:00 PM |
| Prepared by | Scrum master |
| Attendees | Scrum master, Developers, Business stakeholders and QA |

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint status | Things to demo | Quick updates | What’s next |
| 3 out of 4 committed stories are completed | * Real time slot availability module * Venue filtering enhancements * First version of user engagement dashboard | * Bug fixes rolled out for booking overlap * UI improvements applied to filter panel | * Extend report functionality * Implement automation for re-engagement messages |

Meeting type 3: Sprint retrospective meeting

|  |  |
| --- | --- |
| Date | 29/4/2025 |
| Time | 2:00 PM- 3:00 PM |
| Prepared by | Scrum master |
| Attendees | Scrum master, PO, Developers and QA |

Agenda

|  |  |
| --- | --- |
| Topic | Note |
| What went well | Collaborative planning, timely delivery of major stories |
| What didn’t go well | Delay in receiving UI assets for engagement module |
| Questions / feedback | Can we improve sync with design team pre-sprint? |
| Reference | Jira sprint 1 |

Meeting type 4: Daily stand up meeting

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Question | Name/ role | **Sprint Week : (from 15-Apr-2025 to 21-Apr-2025)** | | | | | | |
| Mon | Tue | Wed | Thu | Friday | Sat | Sun |
| What did you do yesterday? | Developer 1 | Story PB-01 | Bug Fix | Code Review | Integration | Story PB-04 |  |  |
| Developer 2 | Setup Filter UI | API work | Linked APIs | Testing | Feedback fixes |  |  |
| What will you do today? | Developer 3 | Dashboard Layout | Metrics Integration | API testing | Document update | Code clean up |  |  |
| Developer 1 | API Optimization | Demo prep | Finalize UI | Refactor | N/A |  |  |
| What (if any) is blocking progress? | Developer 2 | Bug validation | Test Report | Bug Fix | N/A | Planning |  |  |
| Developer 3 | Story Estimation | Sync with QA | Visual Fixes | N/A | None |  |  |
| Developer 1 | None | UI lag | None | None | None |  |  |
|  | Developer 2 | Waiting on backend | None | None | None | None |  |  |
|  | Developer 3 | None | None | Metrics unclear | None | None |  |  |