Meeting minutes

M000 131024 Agenda

Date: 13.10.2024
Time: 12:00
Location: online
Expected Duration: 1 hour

---

1. Welcome and Introductions (5 minutes)

- Welcoming the team and outlining the purpose of the meeting.
- **Everyone is a developer!**

2. Project Overview (10 minutes)

- Arnold presents the project topic: **Online Tutoring Platform**.
- Discuss the main objectives and expected outcomes for the semester.
- Overview of the **key features** (user registration, tutor search, video tutoring, etc.) and Minimum Viable Product (MVP).
- Decision whether we want to stick with the topic?

3. Review SCRUM Process & Roles (10 minutes)

- Brief explanation of SCRUM methodology and how it will be applied.
- Assign team roles:
  - **Scrum Master**: TBD
  - **Development Team**: TBD
- Explanation of each role's responsibilities.
  - **Project Manager**: Oversees project progress, schedules meetings, and ensures deadlines are met. Maintains comprehensive project documentation, from requirements to architecture.
  - **Lead Developer**: Coordinates development efforts, making key technical decisions.
  - **UI/UX Designer**: Focuses on user experience and interface design.
  - **Backend Developer**: Handles the server, database, and logic-heavy components.
  - **Frontend Developer**: Works on client-side elements like interfaces and user interaction.
  - **Quality Assurance (QA) Manager**: Sets testing strategies and ensures the software meets quality standards.
  - **DevOps/Infrastructure Manager**: Sets up development environments, CI/CD, and handles deployments.
- Clarify communication rules and expectations (e.g., transparency, collaboration).

4. Defining Initial Milestones and Deliverables (15 minutes)

- Review the project phases and upcoming deadlines:
  - Vision Document (Week 2)
  - Requirements and Risks (Week 3)
  - Toolchain setup (Week 4)
  - Architecture (Week 5)
- Discuss how to break these into smaller, manageable tasks.
- Ensure that everyone understands the importance of meeting these deadlines.

5. Setting Up Tools (10 minutes)

- Agree on tools for collaboration:
  - **Slack** for communication (already set up).
  - **Google Calendar** for scheduling meetings and deadlines (still to be set up).
  - **Git** for version control.
  - **Google Drive/Markdown for documentation.
- Decide on the platform to track tasks (Trello, Jira, or any other task management tool).

6. Scheduling Recurring Meetings (5 minutes)

- Propose a regular schedule for:
  - **Half-Weekly Standups**: Choose a day and time (15-minute meeting).
  - **Sprint Planning**: At the beginning of each sprint (every 3 weeks).
  - **Sprint Review & Retrospective**: At the end of each sprint (every 3 weeks).
- Make sure everyone can commit to these times by conducting a quick availability check.

7. Establish Time Tracking and Workload Distribution (5 minutes)

- Review the **working hours expectations**.
- Present the Google Sheet for logging hours and tracking effort.
- Agree on how team members will record their time and workload.

8. Open Discussion / Q&A (10 minutes)

- Open the floor for any questions, concerns, or suggestions regarding the project.
- Discuss any initial thoughts on how to tackle the first milestones.

9. Action Items and Next Steps (5 minutes)

- Summarize decisions made during the meeting:
  - Assigning roles.
  - Setting up tools and repositories.
  - Planning the Vision Document for Week 2.
  - Setting up the shared Google Calendar.
- Agree on the immediate next steps each team member needs to take before the next meeting.

10. Closing Remarks (2 minutes)

- a brief wrap-up and ensuring everyone knows what is expected before the next meeting.

M000 131024 Report

Date: 13.10.2024
Time: 12:00
Location: online
Duration: 47 minutes

Absentees

- Mishi Pavani

Decisions

- Acquaint yourself with the vision document, brainstorm user stories individually followed by a group session, and then add your insights to the vision document.
- Organize a poll to schedule a meeting to gather user stories on Tuesday or Monday evening.
- Decide on the Scrum Master position by the end of the day. If no one volunteers, Arnold will take on the role.
- Plan for a weekly stand-up meeting on either Sunday or Monday evening, with the specific time to be determined.
- Adopt a ticket-based approach for working hours.
- Nikolai is tasked with creating the Jira workspace.
- Arnold will set up the GitHub repository.

M001 151024 Agenda

Date: 15.10.2024
Time: 17:20
Location: online
Expected Duration: 1 hour

---

1. Project Concept

1. **Problem Statement**
   - What problem does our platform solve?
2. **Target Audience and Market Position**
   - Who is the target audience for our product, and how does it stand out?
3. **Key Features**
   - What are the key features of our MVP (Minimum Viable Product)?
4. **Stakeholders and Responsibilities**
   - Who are the stakeholders, and what are their roles?
5. **User Environment**
   - What will be the user environment, and how will it impact usage?
6. **Other Product Requirements**
   - What are the non-functional requirements or standards that we need to meet?
7. **Competitive Analysis**
   - How does our platform compare to competitors?
8. **Future Expansion**
   - What are the potential future features after the MVP release?

2. Vision document workload split

- Proposed Allocation of Tasks
- Submission Deadline

M001 151024 Report

Date: 15.10.2024
Time: 17:20
Location: online
Duration: 100 minutes

Absentees

- none

Summary

- The project aims to develop an online tutoring platform for the students at the THU. The motivation is to help students to learn difficult study topics/areas by providing them additional one-on-one support.
- Anyone who is affiliated to the THU (having an email at the 'thu' domain) can be a user/ stakeholder of the application.
- The stakeholders belong to the following categories:
  - **Tutors: Students/faculty at the THU who would like to offer to teach the enrolling students.**
    - A tutor can through his profile, offer to teach a course available from the THU course catalogue. It can be independent of specific study program, e.g, the courses can be from ‘cts', ‘inf’ etc., the level can be bachelors, masters etc. *Question: Can programs which are not directly from the thu course catalogue, but affiliated to to it, be allowed to be offered, e.g, a session on react/flutter/deep learning etc.?*
    - The program can be offered in the preferred language of the tutor to benefit international students who might not be proficient enough in English/German.
    - The program can be offered in 2 modes:
      - on a one-to-one basis with a single student or,
      - as a real-time session to a wider audience.
    - The program conduction mode is primarily video based.
    - The tutor can benefit from monetary contributions by the students if he/she wishes to do so subject to the guidelines set by the product admins.
    - The tutor can publish his/her schedule from which the students can find suitable slots.
  - **Students: Students/faculty at the THU who can enroll for a course/ session offered by any tutor in the tutoring platform**
    - A student can- using his profile- search courses/ tutors where he/she needs additional help.
    - They can have one-to-one appointments scheduled with a tutor or enrol in a session offered by the tutor to a larger audience.
    - The student can rate the tutor.
    - The student can also track the progress of his courses from his profile dashboard.
  - **Application owners/admins:** Product owners/administrators who facilitate, maintain, oversee the application.
- Application features:
  - It should be able for students and tutors affiliated to the THU to create profiles and have access seamless access to the functionalities permitted to them.
  - The application should support/handle:
    - scheduling of sessions/events by tutors-students, managing calendar.
    - high quality video streaming.

        - Segregation and management of user roles/profiles.
        - Functionalities critical to teaching:
            - e.g., Whiteboards
        - Data security and other related aspects.
    - Communication modes:
        - Primarily intended through the chat functionality implemented in the application
        - University emails
        - For sharing files: Integration with Fileserver at the THU?

M001 Proposed split

1. Introduction and Positioning (Person 1)

- **Tasks**: Write the introduction and positioning of the platform. This will include setting the stage for why the platform is needed, and positioning the product in terms of market fit and audience.
- **Guidelines**: Focus on summarizing the purpose of the platform, providing background on the online tutoring industry, and introducing the main goals of the project.

2. Problem Statement (Person 2)

- **Tasks**: Draft the problem statement using the template provided:
  - The problem
  - Who it affects (students, tutors, etc.)
  - The impact of the problem
  - The benefits of solving the problem
- **Guidelines**: Keep the language clear and focused on the specific challenges the platform aims to resolve.

3. Product Position Statement (Person 3)

- **Tasks**: Write the product position statement based on the structure in the document:
  - Target customers (students, tutors)
  - Key benefit (real-time tutoring, easy booking, etc.)
  - Differentiation from competitors (live video, chat integration, etc.)
- **Guidelines**: Highlight the unique value proposition of your platform compared to others.

4. Stakeholder Descriptions (Person 4)

- **Tasks**: Write descriptions of each stakeholder (students, tutors, platform administrators) and their key responsibilities.
- **Guidelines**: Ensure to clearly define each stakeholder's role and what they contribute to or expect from the system.

5. User Environment (Person 5)

- **Tasks**: Describe the working environment of the users (students and tutors), including details on:
  - Number of people involved
  - Task cycles and time spent in each activity (e.g., booking, tutoring sessions)
  - System platforms in use today and future platforms (web, mobile)
  - Any unique constraints (e.g., remote access, mobile usability)
- **Guidelines**: Keep the focus on how users will interact with the platform.

6. Needs and Features (Person 6)

- **Tasks**: List the core needs and corresponding features (MVP and beyond), including:
  - User registration, tutor search, video sessions, chat, progress tracking
- **Guidelines**: Prioritize the features based on importance to the MVP and plan their releases.

7. Other Product Requirements (Person 7)

- **Tasks**: List any hardware, performance, security, or other external requirements the platform will need to meet, including:
  - Compatibility with web and mobile platforms

    - Performance under high usage
    - Security measures for user data
- **Guidelines**: Keep these requirements at a high level, noting any assumptions or external dependencies (e.g., using WebRTC for video).

Review and Finalization (Person 8)

- Once individual sections are completed, Person 8 will consolidate the content, ensuring coherence and making any necessary adjustments with input from the entire team.

M001 Sample Answers

1. Sample Answer: The problem of students struggling to find reliable, accessible tutoring services affects students and educators. The impact of this issue is reduced academic performance and limited access to personalized help. A successful solution would provide a platform for easy access to tutors, real-time support, and personalized progress tracking.

2. Sample Answer: Our platform is targeted at students from high school to university level, as well as independent tutors. The "Online Tutoring Platform" is a web-based service that offers video tutorials, real-time chat, tutor booking systems, and session progress tracking. Unlike other tutoring services, our platform uniquely focuses on integrating live video sessions and real-time communication into a seamless user experience.

3. Sample Answer: The key features include:

- User registration and profile management for both students and tutors.
- A tutor search and booking system based on subjects, availability, and ratings.
- Live video sessions using WebRTC for real-time tutoring.
- A real-time chat feature for communication during and outside of tutoring sessions.
- Session management, with feedback and progress tracking for continuous improvement.

4. Sample Answer: The primary stakeholders include:

- **Students**: Use the platform to find and book tutors, attend sessions, and track progress.
- **Tutors**: Create profiles, provide lessons, manage bookings, and track student progress.
- **Platform Administrators**: Ensure system functionality, handle user support, and monitor progress.

5. Sample Answer: The platform will be web-based, accessible on mobile devices and desktops. Users can book and attend sessions from any location, making it adaptable to remote learning. We must consider different time zones, student schedules, and the need for a simple, intuitive interface to minimize the learning curve.

6. Sample Answer: We need to ensure platform performance under high traffic, robust security for student and tutor data, and compatibility with mobile devices. The platform must also comply with educational standards for progress tracking and feedback reporting.

7. Sample Answer: Unlike competitors, who often focus on pre-recorded video content or limited real-time features, our platform will offer comprehensive live tutoring services and real-time chat, along with robust progress tracking and a feedback system, providing a more interactive learning experience.

8. Sample Answer: Future expansions could include AI-based tutor recommendations, integration with existing learning management systems (LMS), multilingual support, and advanced analytics for tracking long-term student progress.

M002 201024 Agenda

Date: 20.10.2024
Time: 12:00
Location: Online
Expected Duration: 1 hour

---

1. Welcome and Introductions

- Welcome the team.
- Purpose of the meeting: review the REQ (Requirements) document and distribute the work.

---

2. Splitting Work for the REQ Document

- **Key Sections to be Completed**:
  - **Product Backlog** containing a preliminary list prioritized **User Stories** (PDF)
  - **UML Use Case diagram(s)** (PDF)
  - **Lists of non-functional, technical constraints**
  - **Risk list** (identified risks required, priorization, mitigation not required yet)
- **Tasks**:
  - Task 1: Product Backlog and User Stories

    Objective:
    Create a prioritized Product Backlog for the THUtorium project.

    Details:
    - **User Stories:** Brief descriptions of features (e.g., "As a student, I want to search for tutors by expertise.").
    - **Prioritization Criteria:** User impact, business value, and technical feasibility.
    - **Outcome:** Finalize and document the Product Backlog in PDF format.
  - Task 2: UML Use Case Diagram(s) (PDF)

    Objective:
    Develop UML Use Case Diagrams for the THUtorium project.

    Details:
    - **Purpose:** Illustrate interactions between users (students, tutors, administrators) and the system.
    - **Content:** Identify key use cases such as user registration, tutor search, and session scheduling.
    - **Outcome:** Finalize and document diagrams in PDF format.
  - Task 3: List of Non-Functional, Technical Constraints

    Objective:
    Compile a list of non-functional and technical constraints for the THUtorium project.

    Details:
    - **Definition:** Identify system requirements that affect performance, usability, and security.
    - **Examples:** Scalability, security protocols, response time, and accessibility standards.
    - **Outcome:** Create a comprehensive document outlining these constraints
  - Task 4: Risk List

Objective:
Identify and list potential risks for the THUtorium project.

Details:

- **Content:** Identify risks such as technology failures, resource limitations, and scheduling issues.
- **Prioritization:** Rank risks based on likelihood and impact.
- **Outcome:** Document the risk list, prioritizing risks but not requiring mitigation strategies yet.
- Clarify individual responsibilities and expected deliverables for each section.
- Discuss deadlines and intermediate progress checks (preliminary drafts by mid-week).

---

3. Review Professor’s Notes

Discuss key feedback and comments received from the professor during the last meeting.

- Identify the main areas for improvement and clarification.

M002 201024 Report

Date: 20.10.2024
Time: 12:00
Location: online
Duration: 90 minutes

Absentees

- none

Summary

- it was decided to create a pre-version of the project in order to understand what the next steps are, divide the work correctly and based on this create a high-quality REQ until Wednesday offline meeting at 9 am.
- Mishi and Manav will make a few sketches about our UX.
- Nikolai, Gor, and Arnold will consider the app's requirements from several viewpoints (student, tutor, administrator).
- Erind, Jossin, and Anastasiya will take on the scenario from the point of view of the project owner.

M003 231024 Report

Date: 23.10.2024
Time: 9:00
Location: offline
Duration: 3 hours

Absentees

- none

Key Discussion Points and Outcomes:

1. User Story Transformation:

- **Objective**: The team focused on refining user stories into specific, actionable tasks to ensure clarity and focus for each role involved in the project.
- **Scope of Work**: We created distinct tasks from the user stories, such as Trial Classes, File Storage, Certification Verification, etc
- **Personas**: The tasks were tailored to four key personas:
  - **Tutor**: Needs included tools for course and material management, scheduling, and certification verification.
  - **Admin**: Focused on oversight capabilities, user account management, and content moderation.
  - **Student**: Required access to trial classes, scheduling support, and course materials.
  - **Super User/Developer (SU/Dev)**: Handled overarching project infrastructure, ensuring backend processes support each persona’s needs.

2. Category Structuring and Prioritization:

- **Goal**: To streamline task organization, we categorized the refined tasks into overarching categories like: Account Management, Calendar/Scheduling, Verification, Content Management.
- **Priority Levels**: Each category was assigned a priority rating based on importance and impact:
  - **Critical**: Tasks that are essential to the project's core functionality and success.
  - **Moderate**: Important but not project-critical.
  - **Optional**: Additional features that enhance the platform but are not essential.

3. Incremental Goals (November 7 - November 28):

- **Objective**: To define achievable goals and ensure a structured approach to project milestones, we set incremental goals for our first project period.
- **Milestone and Goal Planning**: We outlined the objectives and assigned roles based on the categories established, allowing each team member to focus on specific areas of responsibility.

Additional Notes on Assigned Responsibilities and Risk Management

During the meeting, specific roles and responsibilities were delegated to ensure that all critical elements are managed effectively in alignment with project priorities.

1. **Risk Management**:
   - **Assigned to**: Anastasiya was tasked with creating a comprehensive **Risk Management List**.
2. **Prioritizing User Stories**:

- **Team Collaboration**: The remaining team members were assigned to divide and analyze user stories, to determine the most prioritized tasks to include in the initial **Backlog** list.

M004 271024 Report

Date: 27.10.2024
Time: 10:00
Location: offline
Duration: 2 hours 20 minutes

Absentees

- none

Summary

The team reviewed user stories, which helped us outline tasks in Jira. We then added essential tasks to Sprint 1, which Anastasiya allocated and launched:

- **Arnold:** Set up CI/CD Pipelines, manage version control + documentation, and set up the initial dev and test environment.
- **Erind:** Get familiar with the tech stack for the backend.
- **Mishi:** Create the website’s mock-up design (UI/UX).

Afterward, the team discussed the tools they’d like to use for the project.

Meeting with Professor 17.10.2024

1 Global tutoring platform, not binded to THU, since it should applied to all people, people could provide valid tutoring after finishing the program. We could also think of different university, so tutor can come from anywhere , as well as tutor.

2 Platform will offer additional courses , not binded to the campus st of subject. Main objective - learn and help.

3 Integrate both group and individual

4 Online is more important , but in person also cool (NI)

5 Scheduling - in person and ... register for online tutoring (accept/decline/tentative)

6 WebRTC can be replaced with anything we like. Document only proposes the feature.

7 Main focus : bring together, booking, rating , search, lookup, tutoring, chat

8 Course progress: example is Moodle , Udemy, rely on tutor is only can option and then provide an evaluation.

9 Change Vision to more broader scale , not THU strictly

10 Verification algorithm: Not possible within our project, must be an external algorithm. You could easily think of a checkbox , which has to be done by supervisors of the platform (Admin at deployment). EXTRA MODULE. SUPERVISOR from the quality assurance

11 Security concerns: Since no personal data - nothing needed. But if there is - deal with it.

12 Features review: We cool. Features will be redefined

13 Architect role , quality assurance

14 Text tag : what should we use for such applications : Prof said - use what we can use the best and where we have the most expirience. Nothing special in terms of tech. What you think is the best - use it. High level API’s are cool and welcome.

15 Idea of the project - dive deep into object oriented programming.

16 Try with clean OOL , apply design patterns and improve.

17 In the end he expects a file with a good structure , which provides every deliverable, but code itself. All documentation produced must be there.

18 Next week: 1 Backlog 2 Use case diagrams not really necessary 3 User stories are a MUST 4 Decide what and when wee will cover it. What do we need to achieve for an upcoming increment. 5 One increment = 3 weeks : check calendar 6 Homework: when to meet with the professor. Add Professor to Confluence , JIRA , GIT.

The website should be global, not binded to the University

To have it not worldwide website but the domain within Germany

Don't neglect the “in-person” part of tutoring

Bring people together, register, search and book, online tutoring,

If its online have a chat functionality but not only for when you have an online class

Have progress tracking, did someone actually improve. Maybe have the tutor track progress for students

Software Architect and Project manager

What type of hosting, API’s

Vision:

- **Objective**: Create a global tutoring platform not restricted to any specific university, allowing tutors and students worldwide to connect.
- **Key Features**:
  - Global access: Tutors and students from any location can participate.
  - Focus on learning and helping, with additional courses available beyond traditional academic subjects.
  - Offers both group and individual tutoring sessions, with a strong emphasis on online tutoring but supporting in-person sessions as well.
  - Scheduling system for registering, accepting, declining, or marking tutoring sessions as tentative.

Key Components:

1. **Online Tutoring & In-Person**:
   - Online tutoring as the primary focus, with chat functionality available outside of scheduled sessions.
   - In-person tutoring also supported with appropriate scheduling.
2. **Booking & Rating**:
   - Integration of booking systems for both online and in-person sessions.
   - Rating system for tutors and students to ensure quality.
3. **Progress Tracking**:
   - Tutors can track student progress, with evaluation mechanisms inspired by platforms like Moodle and Udemy.
4. **Verification & Quality Assurance**:
   - External verification algorithms for tutor verification (e.g., supervised by a platform admin).
   - Supervisors from the quality assurance team will handle validations.
5. **Security**:
   - Since no personal data will be collected, security concerns are minimal. If personal data is included, appropriate security measures will be implemented.

Technical Aspects

Architecture & Development:

- **Software Architecture**: Designed to scale globally, integrating both in-person and online tutoring systems.
- **APIs**: High-level APIs should be used for flexibility and ease of development.
- **WebRTC**: May be replaced with another technology; the document only proposes this as a feature.
- **Design Patterns**: Emphasis on clean Object-Oriented Programming (OOP) with the application of design patterns to enhance system architecture.

Roles:

- **Architect**: Responsible for overall project design and ensuring a quality implementation.
- **Quality Assurance**: Ensures that features are properly implemented and verified by platform supervisors.

Project Roadmap

Increment Structure:

- **Increments**: Each increment spans 3 weeks.
- **Backlog & User Stories**: User stories are mandatory for each increment. Focus on defining clear deliverables.
- **Next Steps**:
  a. Establish backlog.
  b. Prepare user stories for upcoming increments.
  c. Plan a meeting with the professor to align expectations.
  d. Add the professor to Confluence, JIRA, and GIT.

Meeting with Professor 24.10.2024

- Decide on tool chain ASAP. Choose tools that align well with our project’s requirements. Consider factors like scalability, team familiarity, and integration with other systems..
- Prepare to show things on Jira/Confluence (if needed) as a backlog process, steps we want to do, steps in progress, etc. Start from where we are.

Meeting with Professor 31.10.2024

Stand-Up Review

- Facilitated by Arnold, the team conducted a regular stand-up meeting.
- Presented the toolchain, initial design concepts, and setup progress to the professor.

Professor’s Feedback and Recommendations (Sprint Planning and Task Definition)

- Clearly plan each sprint by defining all tasks in detail.
- Prioritize user stories in consultation with the product owner.
- Focus on identifying and addressing risky elements at the start of each sprint to minimize potential obstacles.
- Approach risk analysis thoroughly: consider each story in terms of potential pitfalls and ensure early attention to any high-risk areas. (Don’t be too high level with risks, think deeper about what could go wrong. For each story somehow we should figure out if there is anything risky about it and if we should address it ASAP)
- For very first sprint make tasks as definite as possible , more detail - the better. It must be done to mitigate the skill gap between team members.
- Define goals for upcoming sprint, to achieve the goal - this set of tasks must be done. Mid sprint do a review in terms of progresses and if needed drop some tasks , which are optional. By the end of the sprint we want to have something that really works.
- Best way to refine the backlog (detail user stories) = split user stories into different domains and make team (1 experienced 1 not so experienced) and detail them.
- Data model comes after class diagram. DEVELOP IT.

Feedback

- Each one of us should contribute to the meetings. Take over your role and contribute as much as possible.

Meeting with Professor 07.11.2024

- Transfer to UML
- Do we need a class for calls? - discuss
- Dangerous to draw it in the same style
- Is our course to record the progress or is it more like a tutorial - discuss
- Repetition, find a SOLUTION for it
- Rating is also for 1 to 1 meeting
- Think about what do I rate, is it 1 to 1 meeting or 1 to many, 1 to 1 meeting - does it get rating
- **Professor agreed on our model**
- Develop a model further on (UML), all the relationships, the attributes,
- What is a relationship from course to calendar
- On one hand we have students, on the other hand we have requests, the tutor is attached more or less to offers
- In a very central place we have tutor, does it need to be the main thing, in terms of business, if this would be costly, we would have to pay for this service - discuss
- Do we really need these 3 tables (maybe combine them in one), because there are not so many differences, professor told it makes the approach much more complicated
- A typical way is to start by domain model and transfer it to SQL
- The most important thing is think deep when we start with out domain model, what can happen if i do this, how it will work and etc
- Categorization (phd, etc)

Meeting with Professor 28.11.2024

- Architecture Notebook (describing db, front model and etc) asap
- Any diagram requires the description
- What went well and what didn't, lessons learnt for the project doc (start now so it wont be a big deal then)
- Meeting minutes (we have)
- Documentation for backend

Meeting with Professor 05.12.2024

- images where we show our elements and how they are connected + explanation is a good approach
- explaining why (maybe with arguments) we used this or that tool as the description can be found on the page of the tool
- more insight of how you use the tool, in which cases its used, images can really help
- use UML to show the connection between the tools
- lessons learnt