**Reviews of other teams**

**Team D4 Feasibility Study Review**

**Habitab - Rental Property Management System**

**Problem Statement and Summary**

* Concise summary and well-defined problem statement
* Rapid insights on current competitors and what they are doing is also mentioned
* An overview of features given, along with what makes their product standalone
* We found other competitors that they didn’t mention and can take a look at - Metrix-X, Property Vista

**Current Processes**

* A clear explanation of products in the market and their features.
* A view in terms of legacy apps and new technology incorporating apps has also been given
* Target user organization information has also been provided

**System objectives**

* Hassle-free service to both the tenant and the landlord
* Services offered - Chat room→one-to-one, one-to-many, Search, Analytical portal for landlord→Adding New Rental Property, Visualization, Reports Generation, Maintenance and Issue Reporting, Bill Cycle and Rental Remainder, Document store and upload

**Block Diagram**

Well Drawn

**Targeted Customers and Benefits**

* Clearly explained target customers and their benefits

**Technology Considerations**

* A good set of technology considerations

**Product/Service Marketplace**

* Clearly explained the product and service marketplace

**Marketing Strategy**

* Team 4’s marketing strategy is primarily based around ads on social media platforms and hiring influencers on such platforms. They also plan to use referral systems.
* The second important strategy for them is to build a highly accessible and friendly customer support system by sending emails, providing feedback and help infrastructure and so on.
* One important area of improvement for Team 4 can be to focus more on organic ways to gain presence in the market apart from the paid services of ads and influencers.
* They could also work more towards identifying and narrowing down their target audience so that their marketing campaigns would be more efficient and cost-effective.

**Organization, Staffing, and Schedule**

* No staffing problem
* Agile-Scrum Model - Picked well

**Financial Projections**

* Assumes an increase in user base by at least 20% in the first year and 10% annual increase in the subsequent years
* The projection assumes no revenue for the first 5 months of dev and testing
* Ambitious financial projections

**Issues**

* Habitab will initially be released as a web application which might not be enough as a lot of users would want it to be a mobile application because of the ease of use which expects the app to be compatible with multiple mobile operating systems
* There are functionalities which will require storage of files so how will this be done in relational databases
* Security since information related to rental agreements and sensitive documents will be stored.
* Many use cases in the application are based on analytics and NLP, and these algorithms are CPU and memory intensive which might cause a delay in response.

**Assumptions and Constraints**

* Assumed to work 24\*7 once the app is deployed.
* Expected to release by end of November with a full-time operational life
* Expected to have yellow pages data (Either by API calls or scraping)
* Assumed sufficient capital for running servers
* Payments simulated due to security reasons
* Availability and scalability constrained due to small scale local servers
* Constrained to updates due to lack of version control

**Alternatives**

* Extension of the proposed system
* The feature which allows only the landlord and tenant to communicate.
* In case of any maintenance issues or complaints, the tenant will raise the issue in the chat.
* A community management service - not very clear on this aspect
* In this instead of dealing with leases/rents, the main idea here is to bring all the active users of the environment like tenants, landlords, residents and association management people. So this system has an additional hierarchy.

**Queries**

* The usability of this app to each person among the target population can be challenging ( a significant risk factor). But we intend to provide a great user experience and reliable solutions to make it reach through every person in the target population - not very clear, could help with more explanation
* Since Habitab introduces a lot of new technologies how are you thinking about introducing it to new landlords or tenants?
* Alternative 2: Community Management Service -> Manual renewal of lease risk?
* Data collection sources can be clearly explained (Scraping → Security issues)
* Marketing Strategy: further elicitation on how personalization based on customers interest is done is required
* Could you elaborate more on the advanced search options?

**Team D3 - SRS Review**

**Project Title - Learning Portal**

**Purpose and Intended Audience**

* Provided a detailed description of the Web-based system called "Learning Platform".
* Explained the purpose and features of the system
* Interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

**Scope**

* Mentioned all the functionalities with respect to the intended audience and also what the system can’t do.

**Product Perspective**

* Keeps track of user progress point is a little blurry. It doesn’t tell much about how this is being stored in DB. Like what attributes are being used to store the progress?
* Also, there is no mention of storing the faculty and content creator details.
* The other points seem fine considering this section requires things to be written without getting into many details

**Product Features**

* The diagram served its purpose well
* Maybe a few of the features were just part of a bigger functionality

**User Characteristics**

* Well written

**Operating Environment**

* Mentioned technologies used, non-functional requirements and their implementation approach.
* Precise and clear about their idea.
* “The product is expected to be deployed in real-time for better testing and performance.” - could elaborate this a little more and talk about how they are parametrizing “better” testing and “performance”

**Constraints**

* Heavier frameworks - issues while running due to performance limitations of containers -- are you aiming to fix this if it is an issue?
* All containers are Linux based so .NET frameworks can’t run on these -- how are you handling this? Or are you not going to provide .NET frameworks at all?

**Assumptions and Dependencies**

* Users allow cookies to store information only if the website is trusted, so by using SSL certificates, this can be achieved**.** DEPENDENCIES
* Using a free tier AWS account there will be a limit on exceeding that limit there will be a pay-as-you-use policy so how are the limitations handled?

**Functional Requirements**

* Very elaborately explained

**Non-Functional Requirements**

* All in all, the description of non-functional requirements is quite comprehensive and well-thought-out.
* At some places, the requirements could have been stated in a more concise and terse manner.
* The requirements are quite pragmatic and are very likely to be achieved based on the amounts of resources available to the team.
* While 95% availability is a good starting goal, if the system were to grow in business value, the availability would need to increase in order to ensure good user experience on the portal.

**User Interfaces**

* Provide editor preferences
* Switching between multiple language support at any given point of time.
* Different screen modes.
* As your application takes user-defined test cases so an option for that should be provided as the main content in the user interface.
* So according to me, these are also the most important things to be considered in building your application.

**Team D2 - Architecture and Detailed Design Specification**

**Title - Placement Coordination System**

**Architecture design**

* Clear division of components
* a line or two on the accounts component could be written
* application is different from the start that is including the login page or excluding
* It?
* Do you have different logins for students and placement coordinators? If not, how
* do you prevent false positives(students posing as coordinators)? Reminder to
* ensure that extra level of security is present in your application and specified in
* the architectural diagram

**Accounts Application**

* It contains the login page and password reset page to authenticate users and handle any exceptions by allowing the password to be reset.
* not clear if the views are separate from the beginning or only after the verification of the credentials

**Interaction design**

* What do you mean by the default page?
* Eligibility check with the university records or based on details entered by the student?
* Does the company directly contact the shortlist students?
* Revisit border conditions regarding blocking of the user

**Notification**

* How are you planning to test these features as part of your notification system?

**Usage Scenarios:**

* don’t think the actors are themselves usage scenarios, pointing out a naming anomaly

**Architectural Styles and Patterns**

* All architectural styles and patterns are concisely explained of why or why not they are considered.

**Detailed Design Approach**

* clearly mentioned approach.
* also included the concept the design enabling techniques was centered on.
* Included design patterns under all the categories, structural, creational and behavioral, creational design pattern focused on reducing costs for initializing an instance of a class.
* All the included patterns are explained precisely.

**Requirement Traceability Matrix**

Please provides IDs. This section was incomplete.