Vision and Scope Document

for

<EasyTalk>

Version 1.0 approved

Prepared by <ByteBuilders>

< 26.02.2024 >

Business Requirements

Category : Web application for job searching and hiring.

Background

The background involves recognizing the existing gaps in the job market, such as the lack of a centralized and user-friendly platform for job-related communication. EasyTalk aims to simplify and enhance the overall experience for both employers and job seekers by providing a dedicated space that brings together job posting, resume management, and real-time communication.

Main features :

Efficient Hiring Process: By providing a centralized dashboard for application management, EasyTalk streamlines the hiring process, making it more efficient for employers to find the right candidates.

Enhanced Job Seeker Visibility: For job seekers, EasyTalk offers a platform to showcase and manage their resumes effectively, increasing their visibility to potential employers and improving their chances of finding suitable opportunities.

Real-time Communication: The real-time communication feature facilitates quick and effective interaction between employers and candidates, potentially speeding up the decision-making process and reducing the time-to-fill for job vacancies.

Market Demand for Simplified Platforms: With the increasing reliance on digital solutions in the job market, there is a growing demand for platforms that simplify the hiring process. EasyTalk positions itself to meet this demand and become a go-to platform for both employers and job seekers.

Disadvatages :

Customer or Market Needs

Employers: Efficient job posting and candidate management.
Job seekers: Simplified resume submission and easy access to job opportunities.
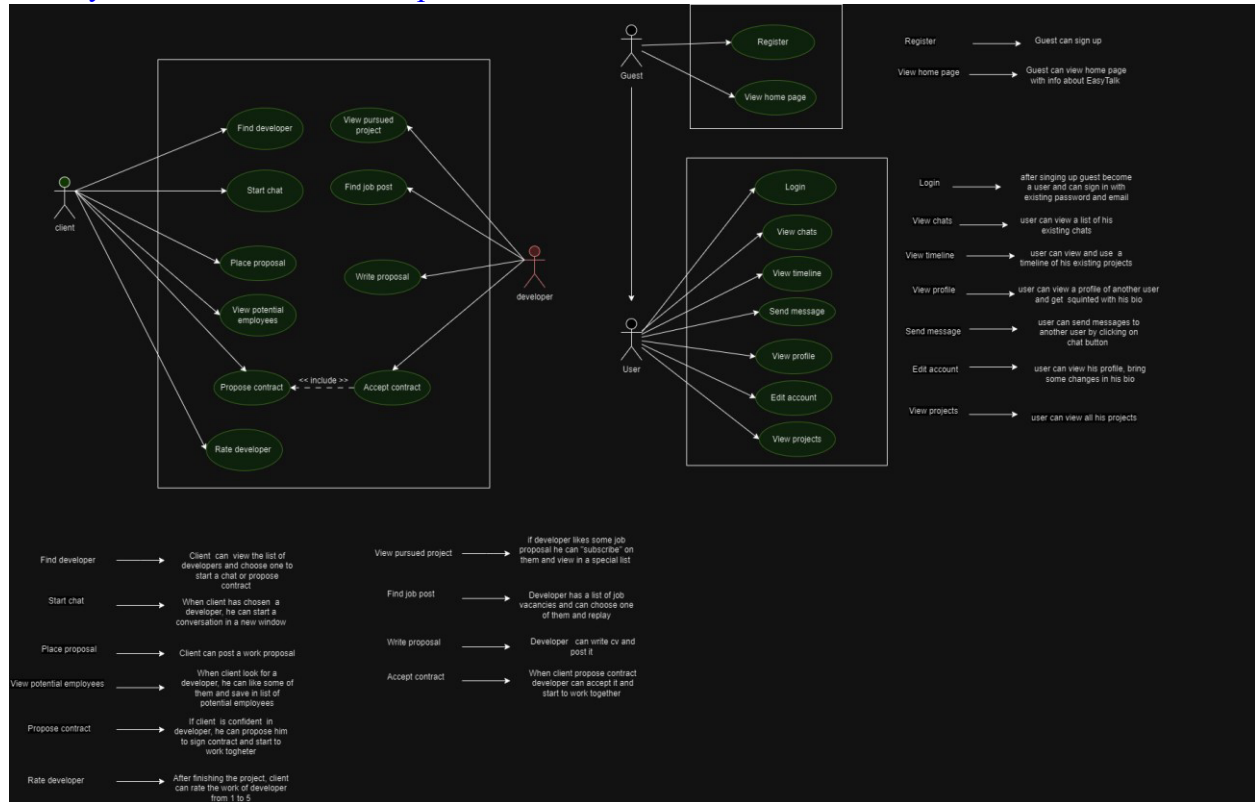*Both parties:* Real-time communication and data visualization for informed decision-making.

Business competitors :
The most popular web applications for job searching are LinkedIn, Faverr , UpWork and etc. Nethertheless our *project is distinguished by the presence of a special timeline for tracking project management. It sets it apart from the rest.*

.

Use-cases

Role	Use Case	Description
Guest	Register	Guest can sign in
	View home page	Guest can view home page with info about EasyTalk
User	Login	After singing up guest become a user and can sign in with existing password and email
	View chats	User can view a list of his existing chats
	View timeline	User can view and use a timeline of his existing projects
	Send message	User can send messages to another user by clicking on chat button
	View profile	User can view a profile of another user and get squinted with his bio
	Edit account	User can view his profile, bring some changes in his bio
	View projects	User can view all his projects

Client	Find developer	Client can view the list of developers and choose one to start a chat or propose contract
	Start chat	When client has chosen a developer, he can start a conversation in a new window
	Place proposal	Client can post a work proposal
	View potential employees	When client look for a developer, he can like some of them and save in list of potential employees
	Propose contract	If client is confident in developer, he can propose him to sign contract and start to work togheter
	Rate developer	After finishing the project, client can rate the work of developer from 1 to 5
Developer	View pursued project	if developer likes some job proposal he can "subscribe" on them and view in a special list
	Find job post	Developer has a list of job vacancies and can choose one of them and replay
	Write proposal	Developer can write cv and post it
	Accept contract	When client propose contract developer can accept it and start to work together

Sequence diagrams

Client sequence diagram

- Client → System: find developer
- System → DB: request developer list
- DB ⇢ System: return developer list
- System ⇢ Client: display developers
- Client → System: start chat
- System → DB: create chat
- System ⇢ Client: display chat view
- Client → System: place proposal
- System → DB: create proposal
- Client → System: view potential employees
- System → DB: request potential employees list
- DB ⇢ System: return potential employees list
- System ⇢ Client: display potential employees
- Client → System: propose contract
- System → DB: save message
- Client → System: rate developer
- System → DB: save developer rate

Freelancer sequence diagram

- freelancer → System: find job post
- System → DB: request job posts list
- DB ⇢ System: return job posts list
- System ⇢ freelancer: display job posts
- freelancer → System: write proposal
- System → DB: create proposal
- freelancer → System: view pursued projects
- System → DB: request pursued projects list
- DB ⇢ System: return pursued projects list
- System ⇢ freelancer: display pursued projects
- freelancer → System: accept contract
- System → DB: save message