Checklist for a simplified coding Turk system

Team:
Members:

Public Information

1. ▢ visitors can **browse** public information made available by clients and developers
2. ▢ visitors can **search** public information made available by clients and developers
3. ▢ Any visitor and user can check the grand statistics of the system, such as number of clients, developers, client(s) with the most projects and developer(s) making most money.
4. ▢ The whole history of any client and developer, including ratings and project details is publicly available

Account Registration

1. a visitor can apply to be a client or developer with deposited money and basic information, a temporary account with applicant's own choice of user id and password is provided—the system should check and ensure the uniqueness of the user id.
   a. ▢ Client account registration
   b. ▢ Developer account registration
2. ▢ the super-user will check the credentials of user applications and decide to accept or reject applicants
   a. ▢ for any rejected cases the super-user must write one sentence describing the reason.

Login

1. ▢ For a **rejected applicant**, when s/he tries to login with the temporary account, s/he will be blocked with reason why s/he was rejected;
2. ▢ For an **accepted applicant**, s/he will be shown a welcome screen and asked to fill in more details about her/himself, such as a resume, picture, interests, sample work (for developer) and business credential (for client), interest, etc., which are available for others to browse

System Demand

1. ▢ a client can post a system demand: including a paragraph describing the system spec and bidding timeline;
2. ▢ a developer can bid on any demand with promised timeline and money.
3. ▢ The client can choose which bidding developer to hire

        a. ▢ if the bidder is not of the lowest price, the client should provide justifications;

Payment

1. ▢ For all money-related transactions, your system provide means for users to put in money and whom to pay/transfer, which can only be finalized by super-user.
2. ▢ Any client with inadequate fund to fulfill a bid will be warned automatically and the posted project nullified
3. ▢ once a developer is chosen, half of the bidding price will be transferred from the client to the winning developer as the front.
4. ▢ If the developer delivered the system before or on the deadline, the remaining half of the price is transferred to the super-user automatically.
5. ▢ For every finished transaction, the system will charge 5%, money transferred to super-user's account, from both the involved client and developer.

Rating

1. The client is asked to evaluate the delivered system with ratings from 1 to 5, 1 being worst, 5 being the best.
    a. ▢ If the rating is >=3, the money held by super-user account is automatically transferred to the developer;
    b. ▢ otherwise, the client should write one note describing why the rating is so low the super-user will discuss with client and developer to decide how much money to give to the developer and back to client.
        i. ▢ Super-user makes the final decision with justifications to change or leave the low rating as is.
2. ▢ The developer can also rate the client after finishing the transaction from 1 to 5, a paragraph must be provided for a rating <=2.
3. ▢ Any developer or client who received average rating is <=2 for >=5 projects is warned due to poor performance;
4. ▢ any user whose average rating to others is <2 (too low) or >4 for >=8 (too high) project is also warned for irresponsible evaluations.
5. ▢ Warned users can send protesting message to the super-user, and the super-user can keep or remove the warning.
6. ▢ Any user who is warned twice is thrown out of the system: s/he can login for one last time for system closing matters then the account is closed forever.
    a. The user can still protest for super-user to reverse this action.

▢ **Creative Feature**