

Metadata

Attendees: Insan, Haoyuan, Igor, Harsha, Peter

Scribe: Peter

Started: 2023-08-11 13:21 +0800

Propose FRACAS Web app

- Story (scenario from user perspective and how will software/feature fit in)
- Cost priority for features
- Deployment (hosting, access via internet or VPN)

Propose FRACAS web app:

Insan: Web application. Accessible via browser/phone.

Insan: Separate database for failure reports in mongodb. Flask backend, react frontend.

Erwin: sent failure report example.

Melinda: be clear about user roles - team lead, creator of report, assigned user.

Erwin: regular member or team lead. Individual report - creator (responsible for dealing with issue in the record - not the same as a reviewer) or owner (person who created report) or both.

Melinda: for example - safety issue => someone else is responsible for dealing with issue, not necessarily person who created the report.

Erwin: anyone can create a record.

Insan: field access => yep.

Melinda: all places have internet access - must be accessible to internet. Look at both exposed to internet or on internal intranet, and decide which option to use later (defer decision until determine pros and cons).

Deployment

Melinda: sort out with the unit coordinator - they have to have a server we can use?

Insan: budget - cheaper than paid options within 10 years. Melinda: just put it towards us. Insan: does your team control UWAM

Erwin: Don't know who controls it in particular.

Melinda: Erwin look into who hosts UWAM. Can't host site - non-trivial. Used to be a server in CSSE for student projects - ask Michael.

Dashboards

Insan: statistics such as ratio of resolved/unresolved, etc. What else to look at in particular?

Erwin: Proportions based on tech teams, proportion of reports unresolved. Anything more complicated requires reliability engineering stuff. Melinda: keep it simple. Number of reports, open, closed, etc. Reports by teams. Easy stats.

Insan <snip> is this an accurate list of teams? Melinda: needs to be documented in the requirements documents.

Melinda: create a new requirements document. with new information. See mockup pages [figma?] as part of proposal. Minimize amount of pages (use dropdowns, etc). Requirements not done until a mockup is done.

Melinda: contact not required.

Erwin: form data - section have to be filled out when creating report. Additional data - can be derived from the application itself or profiles of people. User profile -> contact details already in there.

Melinda: don't add right now (adding new fields for failure report) since we are doing a graph db. Erwin: as a failure recording system, not many other fields required as it is specifically for failures. No team-specific fields.

Insan: Are images required? Erwin: no images for now. Not sure.

Insan: report generation - assume it's an PDF export. Erwin: yes. Not the most critical.

Insan:

Cost priority:

Allocations:

Requirement	Description	Alloc
Auto-completion	The system shall auto-complete as much of the form as possible, considering the failure owner's role in the organisation as well as past records of the owner (allowing changes if incorrect)	
Device Compatibility	The system shall be accessible from computers and mobile devices.	
Commenting	The system shall support comment threads on failures. This is for feedback and collaborative problem-solving.	
Personal Dashboards	The system shall present users with a personalised dashboard upon logging in, showing their unresolved failures, their past resolved failures, their assigned learning failures, and their book-marked failures, as well as a notifications panel.	
Teams Dashboards	The system shall have dashboards displaying unresolved failures, corrective actions for failures needing validation, and analysis results of failure trends and statistics. This will improve teamwide visibility and will enable communication and scheduling.	
Management dashboards	The system shall present team leads and management with a second personalised dashboard showing the active failure records of their team members, as well as a section for failures requiring validation of completeness/accuracy/adequacy.	

Requirement	Description	Alloc
Learning Assignment	The system shall allow management to select past failure records and assign them to members for learning or review. The failure records will be visible on the assignee's personalised dashboard and will be marked as reviewed once read.	
Teams management	All team management features - account management, ownership, etc.	

Melinda: do this by early next week @Erwin.

Deployment (hosting, access via internet or VPN)

Etc

Erwin: how to record comments. log a failure / writing analysis, any other team member should be able to open the record and write a comment. Store as a string. People can comment and it appends to the string (keep it simple). Unstructured comment is the simplest way. Replies would be neat but string is sufficient. Reply function.

Insan: failure resolved but failure appears again. Melinda: new failure report - not re-opening old report.

Melinda: do a flow diagram for user interactions. More detail here -> less programming. Make the user flow as simple as possible. Don't forget about admin view as well (role-based flow diagrams).

action items

Action	Due
Do requirements document + figma + flow diagrams	Wednesday

Meeting adjourned 2023-08-11 1403 +0800