VIRTUAL JUKEBOX - PROJECT BRIEF

BACKGROUND

Jukeboxes were once a common sight at diners and cafés. They offered a means for music lovers to control music outside of their homes before audio technology became portable. The ability to enjoy music together at a café or even in an outdoor setting (e.g., outdoor markets, weddings, parties, etc.) can often bring people together and set the mood for the occasion. The ability to control the music played at venues can increase customer engagement and interaction.

VIRTUAL JUKEBOX- THE PROJECT

You are to develop a solution called Virtual Jukebox where music can be played from a single source be it in an indoor setting (e.g., restaurant, café, etc.) or an outdoor setting through portable speakers (e.g., outdoor markets, etc.). The Virtual Jukebox host can create a playlist of songs that is tailored to suit the setting and taste of the host or event and categorising them according to the artist, genre, year, etc. Guests within a certain radius will be allowed to connect to the Jukebox and view the song that is currently playing, add songs (from the playlist) to the queue, and view songs that are currently in the queue. The application will have an algorithm where a particular song cannot be played consecutively (i.e., back-to-back) or only after a certain hop. The application will allow guests to suggest songs to be added to the playlist by the host. Guests can add songs to the queue based on a credit system. Credits can only be earned after fulfilling a certain criterion (e.g., time-based).

You will design and develop a solution called *Virtual Jukebox*. *Virtual Jukebox* will consist of a minimum of the following components:

1. A website (mobile compatible) that allows for hosts to create playlists, source for songs, and for guests to interact with.
2. A database to store hosts and guest details.
3. A real-time webservice to source for and play songs.
4. An integrated map (Mapworks) to visualise the location of other Virtual Jukeboxes.

Virtual Jukebox will have the following capabilities:

1. General application requirements
    a. The application must be able to indicate where this playlist is being hosted geographically. Only guests within a certain radius (e.g., 200 metres, or configurable) to the host's Virtual Jukebox will be able to connect.
    b. The web application will be able to search for Virtual Jukeboxes from a geographical location e.g., "find Virtual Jukeboxes near me".
    c. The application should have a chat functionality to allow anyone connected to communicate with each other (this should be programmatically moderated for offensive language).
    d. The application should have the ability to shuffle a playlist randomly and prevent songs from playing consecutively and repeatedly.
    e. Guests will have to use credits to request songs to be added to the playlist. A credit system should be setup for guests to earn credits e.g., 1 credit is earned every 1 minute and guests can only accumulate a maximum number of credits.
2. Hosts
    a. Create a playlist of songs for guests to select from.
    b. View and control the song that is currently playing e.g., skip, stop, fast forward, pause, etc.
3. Guests
    a. Connect to the host's Virtual Jukebox at the venue e.g., scan QR Code or enter a unique code.
    b. Pick from a list of songs to add to the playlist queue.
    c. Vote up/ down for a song that is in the queue to bump it up the queue.

d. Request songs to be added to the playlist.
e. Chat with other guests who are connected to the same Virtual Jukebox.

Virtual Jukebox must be designed for efficiency, with a scalable architecture designed for high volumes of data being processed in short timeframes. As the application intends to store personal user information, data privacy and security should be considered.

As with all software engineering projects (SEP), it will be treated by Amristar as an initiative with real-world potential and commercial value. Your group will have a genuine opportunity to determine its success. It is expected that the application will be fit for use at the end of the project, even as a *beta* version. If the outcome is of sufficient quality, Amristar may attempt to commercialise it or use it for marketing purposes.

Software Development Life Cycle (SDLC)

The project will be conducted using a hybrid Waterfall/Agile SDLC. You will first perform technical investigations, including prototyping, that will form the basis for a subsequent requirement specification. This is followed by a detailed design specification, which enables you to think through the use cases and architectural design. For the subsequent development phase, you will translate your requirements into Agile stories. You will then deliver your solution in three well-organised sprints. This approach provides a relatively high guarantee of a successful project outcome while providing you with a broad and beneficial industry experience that includes commonly used processes and artefacts.

Scope and Deliverables

Your assessment items and weightings are provided in a separate attachment. Note that the way you will be graded differs from that of other SEP students, as agreed with your lecturers. Your deliverables include the following:

- A prototype and detailed report which details the outcomes of your investigations into the technologies which you may use (or not) to deliver the solution.
- A project plan developed in MS Project, showing the various stages, deliverables, timelines, and hours spent by each team member. To be updated as required.
- Functional and non-functional requirements, based on your first deliverable and consultation with Amristar. All requirements are assigned weightings which will determine which requirements are addressed first during development.
- A Software Architecture Specification (SAS) design document based on a template we will provide. It will include use cases and process flows and describes in detail how users will interact with *Virtual Jukebox*. It will also include architectural designs to demonstrate how the various components work together.
- Regular Progress reports describing individual effort, team accomplishments, issues, and current and future activities (*template to be provided by Amristar*).
- A functioning implementation of *Virtual Jukebox* that satisfies the main requirements as defined by you. It will be developed in three sprints during the second semester. You are expected to compile detailed technical and developer's documentation.
- A post-implementation evaluation of the developed application which lists the capabilities of your solution, its commercial benefit/viability and lessons learnt.
- Presentations at the end of each semester, summarising your activities and achievements.

For each deliverable, Amristar is willing to review and give feedback on a draft before you hand in the final version. That way, you have the opportunity to learn from mistakes and improve your work. Shortcomings of drafts will not affect your final mark, but your efforts to improve on shortcomings will.

Status Reporting

Each assessment package includes a status report which details achievements to date, current activities, effort expended, upcoming activities and forecasts, and issues (a template will be provided by Amristar). The status report will have a section that details the activities for each team member, specifically their tasks, hours spent, accomplishments since the last reporting period, etc. This report is important.

Project Start-up Activities

You will need to perform research to obtain an idea of the technologies you will be working with, create a basic prototype, and explore which capabilities to focus on. You will have to explore which framework(s) will suit your solution. You should discuss amongst each other who takes on which role(s), although we expect each of you to have some involvement in all aspects, especially coding.

Expectations and Assessment

Amristar provides grades for the deliverables of these projects. The emphasis of SEP projects hosted by Amristar is on proper practice and the correct use of tools and methodologies, in order to prepare you for a career in industry. If you follow proper practice and put in the required effort, you will receive good marks. If you do not, your marks will be reduced, regardless of whether the developed solution works well or not.

A good project plan is important to facilitate equal contribution and inform people of their responsibilities. You will be required to provide a progress report with each package which includes a (self-written) description of what each team member has done, and the hours contributed. If it is found that one or more team members consistently do not work the required hours, we will consult with the University, and there will be impact on their individual mark for the SEP unit. At the same time, the onus is on your team to report it if people are not pulling their weight.

Due to the current COVID-19 restrictions, the first semester of the project will be conducted remotely. This arrangement will be reviewed at the end of semester one. It is your responsibility to coordinate with your team members to allocate sufficient time on the project. We recommend that you spend at least one day a week working together on the project collaboratively. It is expected that you communicate with your project team regularly and proactively.

Your team will be responsible to set up regular weekly (minimum) meetings with Amristar to keep us updated on the progress of the project and if there are any issues that need to be raised. If you are ill and unable to join in the weekly project meeting, you must inform Amristar personally by email and copy in your lecturers – do not get a team mate to pass the message. Your progress report should mention how many hours you missed due to illness, to avoid it looking like you did not work enough hours.

If you find that you are running out of things to do, or are too overloaded, you must communicate with your project team and with us. Although this is an industry project, we can adjust scope to suit the available time.

If there are any issues, please tell us early. It is easier to address them early than to mitigate them afterwards. This is not a university assignment but an industry project; **not all will go to plan but do not panic**.

Resources

Amristar will attempt to provide you with whatever resources you may require. Please do not hesitate to let us know if there is something you need to be able to progress.

- Amristar will provide Wi-Fi access when (if) working on-site.
- We strongly recommend you use a source code repository, e.g., GitHub.
- We can provide you with access to an internet-accessible webserver where your server-side code can run from for demo or testing.

- We will provide you with a JIRA project for managing your tasks and Agile sprints.
- Technical support: if you run into technical roadblocks, let us know. We will put you in touch with the most appropriate person to help you.

Common Mistakes to Avoid

To achieve good marks, please **avoid** this ever-growing list of common mistakes made by project groups in previous years:

- Do not fail to read this document, only to produce a completely wrong scope.
- Do not spend the first three (!) weeks "brainstorming" because you are unsure where to begin, and then run out of time for the first deliverable.
- If in doubt, ask questions. This is what the project is for – we do not expect you to know it all.
- Do not fail to make use of our willingness to review your work – we prefer giving you the opportunity to improve *before* marking rather than after. Not asking for feedback means throwing marks away.
- If you receive a high mark for a deliverable, do not assume this is the norm, and reduce effort for the subsequent deliverable. You will immediately find we can give out low marks too.
- Read our feedback, even if it is long and unpleasant. There is no excuse to make the same mistake two deliverables (or more) in a row.
- If a team member does not pull their weight, do not let the situation drag on to the point where *your* mark is affected. We, nor your lecturer can salvage your mark if you do not report team problems in time. If we need to bring it up with you, it is too late.
- Do not assume you cannot fail SEP, and consistently prioritise other units over it. Amristar invests large amounts of time ($$$) into these projects. Failure to appreciate this can result in your removal off the project, or complete project cancellation.
- Clashes with your employment are not a valid reason not to show up on-site or at project meetings. However important you believe your job is, you cannot expect to get a degree while being exempted from the associated work.
- Do not send in deliverables last-minute, only to miss the deadline because:
    - Your internet dropped out
    - You did not realise the attachment was too large to email
    - The University mail system was down
    - You thought your teammate was going to send it
- Use **Microsoft Office** to write reports, not an obscure Cloud-based editor with poor compatibility, resulting in mangled deliverables which cannot be read.
- Do not copy/paste the same content for different team members into a progress report. We read everything.
- Some things to note before submitting documents:
    - Perform a full spelling and grammar check in Australian English. Yes, use only Australian English – No American English (or worse, a mixture).
    - Even technical documents require grammatically complete sentences, not shorthand.
    - No contractions in formal documents or communications (isn't, can't, won't, etc.).
    - Avoid emotive language.
    - Do not write documents/reports in first-person story-form. Ever.
    - Proofread the document again, and again, and again…! Do not just check the section of the document that you were responsible for. Proofread teammates' work to ensure consistency.
    - Ensure that the document reads well and not a disjointed compilation of individual sections.
    - Remove all comments and tracked changes from final submitted documents, including your supervisor's.
    - Ensure that table of contents is updated.
    - Ensure consistent formatting of paragraphs, text, tables, etc.

    - Ensure final document version number is 1.x onwards – drafts can be 0.x
- We expect old-school formal communication, as a demonstration of respect. It is better to stand out for being overly respectful than inappropriately casual. Do not salute Amristar staff, our clients, or even your lecturer, with “Hey” for any project-related communication.
- Please, no radio-silence at the start of the second semester. You will have your work cut out for you. Decide among your group which on-site day suits and inform us (should we transition to on-site working arrangement). Do not wait for us to ask you to “*please start*”.