**Artefact 01 Project Requirements & Initial Dev Plan**

**Section 1. Project Description**

**1.1 Project Aim and Objectives**

*- The major project aim/goals:*

+ Design and build a prototypical, interactive graph visualizer/editor for creating, exploring, and analysing networks (e.g., knowledge graphs, dependencies, social/interactions).

*- Objectives:*

+ Provide a graph editor that allows users to creat/import graphs from file, add/delete nodes and edges and edit vertex/edge attributes.

+ Offer multiple graph layouts to shoose from, with support for interactive styling.

+ Enable users to apply filter, search and query tools for efficent graph exploration.

+ Implement basic graph analystics, including degree/betweenness/page rank.

+ Support collabaration and sharing features such as adding comment, taking snapshots, sharing links, and exporting files.

+ Include expandable functions such as datasorces and integrations as well as performance and security enhancements using WebGL/LOD rendering and authentication,

*- Benefits:*

+ Provides students with practical experience in handling software according to customer specifications, including project planning, development, execution and deliver the working solution.

+ The first artefact specifically focuses on writing requirements specifications and planning a software development project.

**1.2 User Personas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Characteristics** | **Needs/Expectations** | **How will he/she/they use the system?** |
| Lukas Thode | Age: 30  Gender: Male  Education: a PhD student at BTH, specializing in the efficient and effective use of Generative AI in Software Engineering.  He has full technical skills. With this product, he want to use an app web that focus to use on desktop, because he doesn’t want to see a huge graph on his mobile. It’s not good for his eyes, he said. | Should be simple secure, easy to use, can hander with large graph, share insights with teammates. | - Create/delete, import and export graph, modify nodes and edges, undo/redo, version history.  - Apply layouts and run analystic  - Share and works with eachother |

**Section 2. Software Requirements**

**2.1 Requirements**

The system:

- Must allow users to create/import graphs(JSON), add/delete/edit nodes and edges, undo/redo and version history.

- Must provide some layouts (force-directed, hierarchical), support drag/snap, rule-based styling, zoom/pan, theming, darkmode.

- Must allow users to apply property filters, text search, subgraph extraction, support shortest-path queries, neighborhood expansion.

- Must provide centrality measures: degree/beweenness/PageRank

- Must allow users to comment on graphs, share via links, save snapshots, role-based access, export praph (PNG/SVG/PDF/GraphML)

- Should connectors for Neo4J, live updates.

- Should support role-based access (user/admin), implement login system (it’s ok with hardcode from the beginning).

**2.2 Non-functional Requirements**

- Performance: the system should handle large graphs up to 50k nodes.

- Usability: need to have documentations for user to use the system.

- Compatibility: run on modern web browsers

**Section 3. Backlog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Work item (User Story/Features/Tasks)** | **Estimated Effort (in hours)** | **Priority** | **Planned for Sprint** |
| **F1.** | **Create/import/edit a graph** | **17** | **Must** | **1** |
| **US1.** | **As a user I want to create a graph so I can work with this.** | **6** | **Must** | **1** |
| T1. | Implement to add a node (name of node) | 1 | Must | 1 |
| T2. | Implement to add an edge (direction, weight, source of edge) | 1 | Must | 1 |
| T3. | Implement default layouts (may be force-directed or hybrid degree-based layout) | 2 | Must | 1 |
| T4. | Implement to make nodes and edges become a graph with the default layout | 2 | Must | 1 |
| **US2.** | **As a user I want to import a graph from a JSON file so I can work with this.** | **1** | **Must** | **1** |
| T5. | Implement to import a graph from JSON file. | 1 | Must | 1 |
| **US3.** | **As a user I want to edit a graph so I can model dependencies** | **4** | **Must** | **1** |
| T6. | Implement to delete a node | 1 | Must | 1 |
| T7. | Implement to delete an edge | 1 | Must | 1 |
| T8. | Implement to edit edge attributes (direction, weight, source) | 2 | Must | 1 |
| **US4.** | **As a user, I want to have history versioning (it’s ok with limited versioning) so that I can undo/redo if it do something wrong** | **6** | **Must** | **1** |
| T9. | Implement to have a limited versioning | 2 | Must | 1 |
| T10. | Implement to undo max 5 steps | 2 | Must | 1 |
| T11. | Implement to redo max 5 steps | 2 | Must | 1 |
| **F2.** | **Layouts and styling** | **16** | **Must** | **1** |
| **US5.** | **As a user, I want to apply layouts (force-direcred, hierarchical) so that I have better picture about the graph** | **8** | **Must** | **1** |
| T12. | Implement to change from the default layouts to force-direcred | 4 | Must | 1 |
| T13. | Implement to change from the default layouts to hierarchical (may be need to choose a root) | 4 | Must | 1 |
| **US6.** | **As a user, I want to have some tools such like manual drag, rule-based styling by attributes, zoom, theming, dark mode.** | **8** | **Should** | **1** |
| T14. | Implement manual drag function | 1 | Should | 1 |
| T15. | Implement manual rule-based styling by attributes (colour or shape of nodes and edges) | 3 | Should | 1 |
| T16. | Implement zoom function | 2 | Should | 1 |
| T17. | Implement theming function | 1 | Should | 1 |
| T18. | Implement dark mode function | 1 | Should | 1 |
| **F3.** | **Filter, search and query** | **25** | **Must** | **1+2** |
| **US7.** | **As a user, I want to filter nodes by type or status, so that I can focus only on relevant dependencies** | **5** | **Must** | **1** |
| T19. | Implement filtering logic on node/edge attributes | 2 | Must | 1 |
| T20. | Ensure graph re-reders efficiently when filter changes | 1 | Must | 1 |
| T21. | Add reset/clear filter option | 1 | Must | 1 |
| T22 | Add filter UI | 1 | Must | 1 |
| **US8.** | **As a user, I want to search for a node by name, so that I can quickly locate a specific entity in a large graph** | **3** | **Must** | **2** |
| T22. | Add search bar UI | 1 | Must | 2 |
| T23. | Implement text matching on node labels and attributes | 1 | Must | 2 |
| T24. | Highlight matched nodes in graph view | 1 | Must | 2 |
| **US9.** | **As a user, I want to extract a subgraph around a selected node (e.g., within 2 hops), so that I can analyze its local structure in detail** | **5** | **Must** | **2** |
| T25. | Design extract a subgraph UI | 1 | Must | 2 |
| T26. | Implement logic to generate subgraph from selection node | 2 | Must | 2 |
| T27. | Render subgraph | 1 | Must | 2 |
| T28. | Provide option to go back to full graph | 1 | Must | 2 |
| **US10.** | **As a user, I want to compute the shortest path between two nodes, so that I can identify the minimal dependency chain** | **5** | **Must** | **2** |
| T29. | Build UI to select the start node and the and node | 1 | Must | 2 |
| T30. | Implement shortest-path algorithm | 2 | Must | 2 |
| T31. | Show path length | 1 | Must | 2 |
| T32. | Handle cases where no path exists | 1 | Must | 2 |
| **US11.** | **As a user, I want to expand the neighbors of a node step by step, so that I can progressively explore the graph.** | **7** | **Must** | **2** |
| T33. | Add the option to the node (may be with right click on this node) | 2 | Must | 2 |
| T34. | Implement expansion logic | 2 | Must | 2 |
| T35. | Load subgraph | 1 | Must | 2 |
| T36. | Prevent performance issues by limiting max expansion depth | 2 | Must | 2 |
| **F4** | **Analytics** | **24** | **Must** | **2** |
| **US12.** | **As a user, I want to compute centrality metrics for nodes, so that I can identify the most important nodes in the graph** | **9** | **Must** | **2** |
| T37. | Implemenet degree calculation for all nodes | 3 | Must | 2 |
| T38. | Implement betweenness centrality algorithm | 3 | Must | 2 |
| T39. | Implement PageRank algorithm. | 3 | Must | 2 |
| **US13.** | **As a user, I want to detect communities in the graph, so that I can see clusters of related nodes** | **5** | **Must** | **2** |
| T40. | Implement community detection algorithm | 5 | Must | 2 |
| **US14.** | **As a user, I want to identify connected components, so that I can understand isolated or linked parts of the graph** | **5** | **Must** | **2** |
| T41. | Implemenet connected componenets detection | 5 | Must | 2 |
| **US15.** | **As a user, I want to highlight analytics results in the graph view, so that I can easily see patterns and important nodes** | **5** | **Must** | **2** |
| T42. | Add UI controls to enable/disable analystic highlights | 2 | Must | 2 |
| T43. | Animate or emphasize highlighted nodes/edges for clarity | 3 | Must | 2 |
| **F5** | **Collaboration & Sharing** | **24** | **Must** | **3** |
| **US16.** | **As a user, I want to add annotations or comments on nodes and edges, so that I can leave notes and explanations for myself or other collaborators.** | **6** | **Must** | **3** |
| T44. | Implement UI for adding/editing/removing comments on nodes and edges. | 3 | Must | 3 |
| T45. | Store comments in database | 3 | Must | 3 |
| **U17.** | **As a user, I want to save snapshots of the current graph view, so that I can preserve graph states for future reference or sharing.** | **5** | **Must** | **3** |
| T46. | Capture current graph state | 3 | Must | 3 |
| T47. | Save snapshot to database | 2 | Must | 3 |
| **US18.** | **As a user, I want to generate shareable links to my graph, so that others can view or collaborate on it without needing direct access to my system.** | **3** | **Must** | **3** |
| T48. | Implement to generate unique shareable URLs. | 3 | Must | 3 |
| **US19.** | **As a user, I want to assign roles (viewer/editor) to collaborators, so that I can control who can modify the graph and who can only view it.** | **6** | **Must** | **3** |
| T49. | Define user roles and permissions | 3 | Must | 3 |
| T50. | Implement access control check | 3 | Must | 3 |
| **US20.** | **As a user, I want to export the graph as PNG, SVG, PDF, or GraphML, so that I can use it in reports, presentations, or other tools.** | **4** | **Must** | **3** |
| T51. | Implement PNG export | 1 | Must | 3 |
| T52. | Implement PDF export | 1 | Must | 3 |
| T53. | Implement GraphML export | 1 | Must | 3 |
| T54. | Implement SVG export | 1 | Must | 3 |
| **F6** | **Data sources & integrations** | **15** | **Must** | **3** |
| **US21.** | **As a user, I want to connect my graph editor to Neo4j so that I can import and visualize my data seamlessly.** | **5** | **Must** | **3** |
| T55. | Implement backend API routes to connect to Neo4j | 5 | Must | 3 |
| **US22.** | **As a user, I want the graph to automatically update when the underlying data changes, so that I can always see the most current state of my data** | **5** | **Must** | **3** |
| T56. | Implement WebSocket or SSE (Server-Sent Events) for live updates | 5 | Must | 3 |
| **US23.** | **As a user, I want to perform bulk imports using ETL pipelines, so that I can efficiently load large datasets into the graph editor without performance issues.** | **5** | **Must** | **3** |
| T57. | Implement import pipeline that parses JSON files | 3 | Must | 3 |
| T58 | Map imported data to graph nodes and edges | 2 | Must | 3 |
| **F7** | **Performance & Security** | **21** |  | **3** |
| **US24** | **As a user, I want the graph to render smoothly even with large datasets, so that I can explore and interact without lag** | **6** | **Must** | **3** |
| T59. | Integrate WebGL-based rendering for graph visualization | 3 | Must | 3 |
| T60. | Implement Level-of-Detail (LOD) techniques | 3 | Must | 3 |
| **US25** | **As a user, I want only parts of the graph to load as needed and reuse cached data, so that performance remains optimal and memory usage is reduced.** | **3** | **Must** | **3** |
| T61 | Implement lazy loading | 3 | Must | 3 |
| **US26** | **As a user, I want to securely log in to the graph editor, so that only authorized users can access the system** | **3** | **Should** | **3** |
| T62. | Implement user login | 3 | Should | 3 |
| **US27** | **As a user, I want to have role-based permissions, so that I can control who can view or edit graphs** | **3** | **Should** | **3** |
| T63. | Implement backend permission checks for API operations | 3 | Should | 3 |
| **US28** | **As an admin, I want to track user actions in the system, so that I can monitor changes and ensure accountability.** | **6** | **Should** | **3** |
| T64. | Track all significant user actions (graph edits, imports, exports, logins). | 2 | Should | 3 |
| T65. | Store logs securely with timestamps and user IDs | 2 | Should | 3 |
| T66. | Provide admin UI to view/search logs | 2 | Should | 3 |

Kraven: hur snabbt det var

Risker: en tabell

Kolla med att impotera filen på databasen => kunden kan använda dem igen. Type auto-save

Weekly status: tänker på varför