# Risks

1. Bibliotheken /Frameworks und die Probleme die auftreten können
   1. Die JavaScript-Bibliothek ist für uns zu kompliziert um sie effektiv einzusetzen
   2. Das Datenformat das die ausgewählte Bibliothek verwendet ist unpraktisch oder sehr kompliziert
   3. Die Bibliothek ist zu simpel bzw. zu beschränkt um die Aufgabe zu erfüllen
   4. Die Bibliothek kann die riesigen Datenmengen aus Wikipedia nicht effektiv aufnehmen und darstellen
2. Das Endprodukt erfüllt nicht die Standards und Ziele die wir uns gesetzt haben
3. Wichtige Dateien warden beschädigt
4. Das GUI wird nicht rechtzeitig fertig wegen
5. Gesundheitlichen Gründen
6. Schlechter Aufgabenteilung
7. Einem Computerausfall
8. Die Datenmenge die Wikipedia ausspuckt ist zu gigantisch um (flüssige Funktionsfähigkeit des GUIs bei Seite gelassen) diese überhaupt effektiv darzustellen
9. Keiner im Team GUI hat viel Erfahrung mit HTML, JavaScript, SVG etc.

# Measures to take to minimise the risks

1. Mehrere Bibliotheken / Frameworks sollten gesichtet werden, damit man ihre Vor- und Nachteile vergleichen und sich schlussendlich auf eines festlegen kann.
2. Gute Planung, Ressourcenteilung und Zeitmanagement kann hier den entscheidenden Unterschied machen der über die Qualität des Produkts entscheidet.
3. Es sollte viel gespeichert und vor allem viel in das Git gepusht werden. Dadurch kann bei fehlerhaften Daten leicht auf eine vorherige Version reversiert werden ohne, dass all zu viel Fortschritt verloren geht.
4. Gute Planung kann hier vieles verhindern.
5. Kranke sollten sich rechtzeitig melden und von den anderen auf dem aktuellen Stand gehalten werden. Des Weiteren sollte immer jemand bereit stehen die Stelle des Kranken zu ersetzen. Deswegen ist gute Dokumentation des Codes maßgeblich erforderlich.
6. Wenn die Aufgaben schlecht verteilt sind sollte dies sofort den anderen mitgeteilt werden, damit eine effektive Aufgabenneuverteilung stattfinden kann. Außerdem sollten Teammitglieder die bereits mit ihren Aufgaben fertig sind dies melden und sich entsprechend neue Aufgaben suchen.
7. Ähnlich wie bei 3. sollte möglichst oft ins Git gepusht werden damit bei einem Computerausfall möglichst wenige Daten verloren gehen. Das Problem des Fehlenden Arbeitsgeräts muss Situationsspezifisch gelöst werden.
8. Die Daten können anhand mehrerer Faktoren gefiltert werden bevor sie dargestellt werden um die Anzahl der einzelnen Noden zu reduzieren und spezifischere Ergebnisse zu bekommen.
9. Das Internet kann diesen Teammitgliedern auf die Sprünge helfen. Bei weiteren Fragen können sie sich auch an die Teammitglieder wenden die etwas mehr Erfahrung mit den benötigten Programmiersprachen besitzen.