1 PROJEKTBESCHREIBUNG

* 1. PROJEKTUMFELD  
     Das Projekt wurde von uns neben der Berufsschule teils auch in den Räumen der Netzkern AG in 42285 Wuppertal geplant und erstellt.  
     Unser fiktiver Kunde hat sich eine Extension gewünscht, die es ermöglicht Wörter in eine Fremdsprache zu übersetzen und die Sucheinträge so zu speichern, dass diese in einem späteren Zeitpunkt als eine Art Vokabeltrainer wieder abgerufen werden können.  
     Wir haben diesen Auftrag angenommen.
  2. PROJEKTZIEL  
     Im Webbrowser „Chrome“ von Google kann der Endbenutzer die Extension installieren, wenn er auf den von Google zur Verfügung gestellten „Chome Web Store“ geht und hier nach den Namen der Extension sucht. Die Applikation wird automatisch in den Chrome Browser installiert und kann sofort angewendet werden.  
       
     Der User muss sich zuvor beim Anbieter der Applikation registrieren damit seine Suchergebnisse gespeichert und wieder verwendet werden können. Ist der User angemeldet, kann er jetzt Wörter in das Eingabefeld eingeben und es anhand eines Menüs in diverse Sprachen übersetzen lassen. Während des Suchauftrages wird neben dem gesuchten Wort auch das übersetzte Wort in eine Datenbank als eine Art Karteikarte gespeichert. Um Redundanz beim Anlegen von Karteikarten zu vermeiden, erhält jede Karteikarte eine ID anhand derer festgestellt werden kann ob diese Karteikarte bereits in der Datenbank existiert und demnach aussortiert werden.  
       
     Hat der User eine gewisse Anzahl von Suchaufträgen gestartet und existierende Wörter übersetzen lassen, kann er sich diese in Form eines Karteikartensystems abfragen und prüfen lassen.
  3. PROJEKTSCHNITTSTELLEN  
     Der Endbenutzer kann die Extension in einem Google Chrome Browser benutzen.  
       
     Der Webserver basiert auf der Model View Controller (MVC) Technologie von ASP.Net.  
     Das Frontend ist mit Html 5, JavaScript und Css 3 entwickelt.  
     Die MySQL Datenbank liegt auf einem Microsoft SQL Server.

1. PROJEKTPLANUNG  
   1. PROJEKTPHASEN  
      Das Projekt benaspruchte einen Aufwand von 70 Stunden. Eine grobe als auch detaillierte Zeitplanung finden sich im Anhang unter A.1 Grobe Zeitplanung bzw. A.2 Detaillierte Zeitplanung.
   2. ABWEICHUNGEN VOM PROJEKTANTRAG  
      Es gab keine Abweichungen vom offiziellen Projektantrag
   3. RESSOURCENPLANUNG  
      Die benötigten Ressourcen sowie die verwendete Hard- Und Software finden Sie im Anhang unter A.3 Ressoucenplan und verwendete Hard- und Software.
   4. ENTWICKLUNGSPROZESS  
      Als Entwicklungsprozess wurde die agile Softwareentwicklung verwendet, sodass wärend der Implementierung ständige Rücksprache mit der Geschäftsführung, dem Kunden und dem Leiter der Entwicklungsabteilung bestand.

2.5 DAILYPLANNING  
Wir haben entschieden auf Grund der größe und Aufwand des Projekts anstatt Dailys Weeklys zu nutzen um einen Austausch von Informationen eines Sprints zu gewährleisten.

1. PRODUKTBACKLOG  
     
   3.1 ANFORDERUNGEN  
   Es soll eine Extention nur für den Google Chrome Browser sein.  
   Sie soll eine API zur Übersetzung benutzen und alle Suchergebnisse in eine MySQL Datenbank abspeichern.  
     
   3.2 AUFGABEN  
   Zu Beginn soll ein ER Modell erstellt werden um die Datenbank zu planen. Gefolgt sollen Programmablaufpläne erstellt werden die einen allgemeinen Überblick über die Architektur der Software darstellen sollen.  
   Es wird das Visual Studio 2013 / 2015 benutzt um eine komfortable Möglichkeit zu haben mit dem Entity Framework zu arbeiten.  
     
     
     
     
     
     
     
     
     
     
   Hier beginnt die Erstellung der Grundarchitektur der Extension und die Implementierung der API zur Übersetzung.  
     
   Die Datenbank wird aus dem ER Modell in eine SQL Datenbank übertragen. In dieser Datenbank werden Karteikarten und Karteikarten Sessions angelegt und eine Userverwaltung angelegt.  
   Es muss das Entity Framework aufgesetzt werden und die Local Storage.  
   Jetzt folgt die Validierung der Eingaben die überprüft ob das gesuchte Wort existiert und auch richig geschrieben wurde.  
     
   Steht das Backend und die Datenbank, wird mit dem Design begonnen. Es soll ein User Interface geben dass die Ein und Ausgabe der Suchaufträge im Browserfenster grafisch darstellt.  
     
   Ist die UI fertig gestellt, wird das ganze Produkt auf Fehler überprüft und ggf. behoben.
2. ANALYSE

4.1 IST-ANALYSE  
Extensions sind heutzutage ein fester Bestandteil eines jeden Browsers, die es ermöglichen dem Benutzer ein individuell auf sein „Surfverhalten“ angepassten Browser zu konfigurieren.  
  
Nach einer Installation und Aktivierung der Extension im Browser kann der User über ein Eingabefeld Wörter übersetzen lassen. Sind 5 Wörter gesucht worden, wird man darüber informiert, dass er eine Lern-Session starten kann. Nimmt man Sie nicht Anspruch, wird der Erinnerungscounter automatisch auf 5 erhöht. Um diesen Service nutzen zu können muss sich der User spätestens jetzt registrieren.  
  
Startet man die Lern-Session, werden alle in der Vergangenheit aufgetragen Suchwörter in der Suchreihenfolge wie bei einer Art Vokabeltest abgefragt.

1. SPRINTS I  
     
   4.1 PLANNING I  
   In diesem Sprint starten wir mit dem aufsetzen der Solution und der Vorbereitung der Architektur. Dazu bauen wir die Übersetzungs API ein und testen ob diese reibungslos funktioniert.  
   Zudem erstellen wir ein ER-Modell und setzen danach die Datenbank auf.  
     
     
     
     
     
     
     
     
     
     
   4.2 PLANNING II  
   - ASP.Net MVC Projekt aufsetzen  
   - Architektur der Extension vorbereiten  
   - API implementieren  
   - API Test  
   - ER- Modell erstellen  
   - Datenbank anhand des ER- Models aufbauen  
     
   4.3 SPRINTBACKLOG  
   - ASP.Net MVC Projekt aufsetzen (1 PT)  
   Das Projekt wird als MVC Projekt erstellt und die Ordnerstruktur und Konfigurationen werden an das Projekt angepasst.  
     
   - Architektur der Extension vorbereiten (3 PT)  
   Die grundsätzliche Logik zwischen der Extension und der Datenbank wird vorbereitet.  
   Models werden erstellt.  
   View wird erstellt.  
   Controller werden erstellt.  
     
   - API implementieren (1 PT)  
   API muss implementiert und konfiguriert werden.  
     
   - API Test (1 PT)  
   Funktionstest in alle Richtungen ob die API funktioniert  
     
   - ER- Modell erstellen (1 PT)  
   Ein ER- Modell erstellen mit Visio  
   Überprüfung ob die damit die Datenbank funktionieren kann.  
     
   - Datenbank anhand des ER- Models aufbauen (2 PT)  
   Datenbank Struktur überprüfen  
   Datenbank Mock Tests  
     
   4.4 WEEKLY MEETING  
   Mittwoch 12:00 – 12:15 Uhr (Standup Meeting)  
   In dem Meeting stellte sich raus, dass wir unsere Schätzungen zu optimistisch geschätzt hatten und wir unser Ziel nicht geschafft haben. Die Datenbank wurde nicht ganz fertig und die Mock Tests werden in den nächsten Sprint gezogen.  
     
   4.5 REVIEW  
   Der aktuelle Stand wurde dem Projekt Owner vorgestellt. Er war er sehr überrascht wie schnell wir waren und wie gut das ganze schon funktionierte. Es gab keine Bugs zu diesem Zeitpunkt.  
     
     
     
     
     
   4.6 RETROSPEKTIVE  
   Die Stimmung im allgemeinen war positiv. Es wurde nochmal diskutiert, wie wir die nächste Schätzung im neuen Sprint angehen und uns gegebenfalls etwas zurück halten mit dem schätzen im nächsten Sprint.  
   Dadurch dass wir die Datenbank in diesem Sprint nicht mehr einbinden konnten werden wir uns wohl grundsätzlich im nächsten Sprint mit der Datenbank aufhalten.
2. SPRINTS II  
     
   5.1 PLANNING I  
   In diesem Sprint schließen wir als erstes die Aufgaben mit der Datenbank ab die wir im ersten Sprint nicht geschafft haben. Zudem ist geplant das Entity Framework aufzusetzen dass es uns ermöglicht die Datenbank schneller zu befüllen.  
   Zudem sollen jetzt Karteikarten angelegt werden können und diese im Local Storage gespeichert werden können.  
   Am Ende des Sprints werden wir die Validierung implementieren.  
     
   5.2 PLANNING II  
   - Entity Framework aufsetzen  
   - Datenbank anhand des ER- Models aufbauen  
   - Karteikarten anlegen  
   - Local Storage als Speichermöglichkeit nutzen  
   - Validierung implementieren  
     
   5.3 SPRINTBACKLOG  
   - Entity Framework aufsetzen(3PT)  
   Einarbeitung in Entity Framework (In Zukunft EF).  
   Vorbereitung des Entity Framework und Konfiguration für die Solution.  
     
   - Datenbank anhand des ER- Models aufbauen (1 PT)  
   Mit EF die Datenbankstruktur erstellen  
   Datenbank Struktur überprüfen  
     
   - Karteikarten anlegen (3 PT)  
   Mit JavaScript und Ajax die DB und WEB Request Logik entwickeln.  
   Die Datenbank mit Karteikarten befüllen.  
   Testing.  
     
   - Local Storage als Speichermöglichkeit nutzen (1 PT)  
   JavaScript Logik erstellen um Daten lokal in einer Json Datei zu sichern.  
   Testing.  
     
     
     
     
     
     
     
   - Validierung implementieren (1 PT)  
   Mit HTML eine localside Validation erstellen.  
   Testing.  
     
     
   5.4 WEEKLY MEETING  
   Mittwoch 12:00 – 12:15 Uhr (Standup Meeting)  
   Das Meeting ergab das wir alle Tasks wie geplant erledigen konnten.  
     
   5.5 REVIEW  
   Der aktuelle Stand wurde dem Projekt Owner vorgestellt. Die Tests ergaben keine Fehler.  
   Es wurde vorgeschlagen das Design und den Teil der Abfrage mit mehr Farbe im Design zu befüllen. Die allgemeine Funktion der Extension wurde sehr positiv aufgenommen.  
   Wir sagten dem Designvorschlag zu, da das aktuelle Design rein für Testzwecke bereit stand und wir im nächsten Sprint das Frontend und Design angehen werden und dadurch keinen Zeitverlust haben.  
     
   5.6 RETROSPEKTIVE  
   Das Retro ging gut von der Hand. Wir waren alle zufrieden mit dem aktuellen Stand der Extension und auch dass wir die Schätzungen in diesem Sprint gut bemessen hatten.
3. SPRINTS III  
     
   6.1 PLANNING I  
   In diesem Sprint werden wir das Design und Frontend fertig stellen.  
   Wenn das Design steht wird die Solution komplett getestet.  
   Sind keine Fehler mehr ausfindig zu machen, kann die Solution an den Kunden gehen.  
     
   6.2 PLANNING II  
   - Das Grundgerüst der UI erstellen  
   - Der Abfragebereich wird mit farbigen Hinweisen für die Testergebnisse bestückt  
   - Großes Testing und Bug fixing  
     
     
   6.3 SPRINTBACKLOG  
   - Das Grundgerüs der UI erstellen  
   Mit HTML Forms wird ein UI erstellt. Das UI wird mit Logik und Content aus dem Backend bestückt.  
   Die Abfrage UI wird erstellt und mit Abfrage Logik aus dem Backend bestückt.  
   Zum Schluss mit Mock Content befüllen und testen.  
     
     
     
     
     
     
     
   - Der Abfragebereich wird mit farbigen Hinweisen für die Testergebnisse bestückt  
   Mit CSS 3 und JavaScript wird die Logik in das vorhandene UI eingebunden.  
   Testing der Logic und Frontend  
     
   - Großes Testing und Bug fixing  
   Alle Bereiche des Frontend und Backend werden nochmal getestet.  
   Treten Fehler auf werden diese direkt gefixt.  
     
     
   6.4 WEEKLY MEETING  
   Mittwoch 12:00 – 12:15 Uhr (Standup Meeting)  
   Das Meeting ergab das wir alle Tasks wie geplant erledigen konnten.  
     
     
   6.5 REVIEW  
   Die Extension wurde dem Kunden vorgestellt. Er war begeistert und sehr zufrieden mit dem Ergebnis. Ein Abschluss des Projektes wurde genehmigt. Das Projekt wurde dem Kunden übergeben.  
     
   6.6 RETROSPEKTIVE  
   Das Retro ergab, dass wir alle zufrieden mit dem Ablauf des Sprints waren. Es gab beim großen Test kaum Fehler zu beheben. Das wir den nachträglichen Kundenwunsch verwirklichen konnten ohne auf Probleme zu stoßen, zeigte unsere gute Organisation des Projektes.
4. BEURTEILUNG DES VORGEHENSMODELLS  
   Auf Grund der Größe des Projektes empfanden wir das Modell als zu Aufwendig in der Planungsphase. Der einzige Vorteil den wir hatten, war die Möglichkeit das Projekt in den Sprintabschnitten flexibel zu erweitern.
5. ERKLÄRUNG  
     
   Wir versichern, dass wir den Abschlussbericht selbständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und diejenigen Stellen des Berichts mit genauen Anabe der Quelle kenntlich gemacht haben, die wir im Wortlaut oder im wesentlichen Inhalten anderen Werken entnommen haben.

Datum:

Unterschrift (Andreas Krahl, Patrick Ribbach, Tommy Pototzki):