Projektdokumentation

Beaver (Arpagaus, Petit-Sirigu, Akdemir, Egloff)

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
| 27.10.21 | 0.0.1 | Erste Version | Arpagaus |
| 27.10.21 | 0.0.2 | IPE | Arpagaus |
| 03.11.21 | 0.0.3 | PE | Arpagaus |
| 10.11.21 | 0.0.4 | ER | Arpagaus |
| 17.11.21 | 1.0.0 | Finale Version | Arpagaus |

# Informieren

## Ihr Projekt

Projekt: Kursanmeldung

Durch eine Abstimmung und die Mehrheit stimmt für Kursanmeldung. Es werden Workshops angeboten, an denen sich Personen anmelden können. Die Personen schreiben ihre drei Lieblingsworkshops auf und vergeben Prioritäten: 0 für den liebsten, 1 für den zweitliebsten und 2 für den drittliebsten Workshop.

## Quellen

Unsere Gruppe hat keine Quellen benötigt.

## Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Muss | F | Das Kurseinteilungsprogramm muss alle verfügbaren Kurse anzeigen. |
| 2 | Muss | F | Das Kurseinteilungsprogramm muss den Benutzer nach dem Vor- und Nachnamen abfragen |
| 3 | Muss | F | Das Kurseinteilungsprogramm muss eine gültige Mail Adresse Identifizieren |
| 4 | Muss | F | Das Kurseinteilungsprogramm muss die Kursnamen überprüfen |
| 5 | Muss | Q | Das Kurseinteilungsprogramm muss den Benutzer nach weiteren Eingaben fragen |
| 6 | Muss | F | Das Kurseinteilungsprogramm muss einen Algorithmus haben für die Kurseinteilung. |
| 7 | Muss | F | Kurseinteilungsprogramm zeigt die Einteilung an (Enter) |
| 8 | Muss | F | Das Kurseinteilungsprogramm muss die Vorlieben der Kurseingeteilten einhalten. |
| 9 | Kann | Q | Das Kurseinteilungsprogramm darf nicht weniger als 4 und nicht mehr als 20 in einem Kurs erlauben |
| 10 | Muss | F | Das Kurseinteillungsprogramm muss mehreren Personen erlauben sich anzumelden |
| 11 | Kann | Q | Das Kurseinteillungsprogramm kann fehleingaben umgehen |
| 12 | Kann | Q | Das Kurseinteilungsprogramm kann Dateien mit Kursen erstellen und auch im Programm anwenden. |

## 1.4 Diagramme

## 

## 

## 

## 1.5 Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Vorbereitung** | **Eingabe** | **Erwartete Ausgabe** |
| 1.1 | Programm starten |  | Aufgelistete Kurse |
| 2.1 | "Bitte Vorname eingeben" | Hans | "Bitte Nachnamen eingeben" |
| 2.2 | 2.1/ " Bitte Nachnamen eingeben" | Peter | "Hans.Peter@stud.bbbaden.ch" |
| 3.1 |  |  |  |
| 4.1 | "Geben sie ihre Kursprioritäten ein" | Fussball 0  Basketball 1  Volleyball 2 | Falls nicht korrekte Eingabe "Bitte geben sie den Kurs + Prioritäten ein" |
| 5.1 | "Wollen sie eine weitere Mail Adresse eingeben?" | "Ja" oder "Nein" | 2.1 oder 7.1 |
| 6.1 | 7.1 Kurseinteilung ist zufriedenstellend |  |  |
| 7.1 | Drücken sie eine Beliebige taste, um die Kurseinteilung anzusehen | Enter | "Hans.Peter@stud.bbbaden.ch Volleyball" |
| 8.1 | 7.1 | 1. Fussball, 2. Volleyball, 3. Musik | Vor und Nachname, Tanzen |
| 9.1 | 8.1 |  | Wollen sie eine weitere Person anmelden? (j/n) |
| 10.1 | 9.1 |  |  |
| 11.1 |  | "x" | "Falsche Eingabe, nur j/n" |
| 12.1 | "Wollen sie diese Liste Exportieren? (j/n)"  "j" | "Geben sie einen Speicherort für die Datei ein:" | "C/users/… |

1. Planen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Einteilung** | **Beschreibung** | | **Zeit (geplant)** |
| 1.1 | 10.11 | Petit-Sirigu | Das Programm begrüsst den Benutzer und sagt was er/sie zu tun hat | | 10 |
| 2.1 | 10.11 | Akdemir | Das Programm fragt den Benutzer nach seinem Namen | | 20 |
| 3.1 | 10.11 | Akdemir | Der Benutzer kann seinen Vor- und Nachnamen eingeben | | 10 |
| 4.1 | 10.11 | Akdemir | Das Programm erstellt aus den Namen von selbst eine E-Mailadresse | | 10 |
| 5.1 | 10.11 | Petit-Sirigu | Alle Workshops werden zur Auswahl angezeigt | | 20 |
| 6.1 | 10.11 | Petit-Sirigu | Das Programm fragt in welchen Workshop man sich einschreiben will und dass man als erstes sein liebster Workshop schreiben soll. | | 10 |
| 7.1 | 10.11 | Petit-Sirigu | Der Benutzer kann seine drei Auswahlen in die Konsole schreiben | | 10 |
| 8.1 | 10.11 | Petit-Sirigu | Das Programm merkt sich die drei Auswahlen des Benutzers | | 10 |
| 9.1 | 10.11 | Petit-Sirigu | Das Programm fragt, ob der Benutzer noch weitere Personen anmelden will. | | 10 |
| 10.1 | 10.11 | Petit-Sirigu | Der Benutzer kann weitere Namen angeben und drei Workshops pro Person wählen | | 20 |
| 11.1 | 10.11 | Arpagaus | Falsche Namen und Workshopnamen werden abgefangen | | 20 |
| 12.1 | 10.11 | Arpagaus | Verbesserungsvorschläge werden rot angezeigt | | 30 |
| 13.1 | 10.11 | Egloff | Es wird eine Liste aller Personen und deren gewählten Workshop ausgegeben | | 10 |
| 14.1 | 10.11 | Egloff | Das Programm fragt dann, ob die Workshopeinteilung starten soll. | | 10 |
| 15.1 | 10.11 | Egloff | Das Programm teil automatisch alle Personen in ihren liebsten Workshop ein | | 50 |
| 16.1 | 10.11 | Egloff | In einem Workshop hat es nur 4 – 20 Plätze frei! | | 30 |
| 17.1 | 10.11 | Egloff | Sobald ein Workshop voll ist, werden die nachfolgenden Personen in ihren zweit liebsten Workshop eingeteilt | | 30 |
|  | | | 330 |

1. Entscheiden

Wie benennen wir unsere Workshops? Lösung: Tanzen, Fussball, Volleyball, Musik, Zeichnen, Basketball.

Wie viele Workshops verwenden wir? Lösung: 5

1. Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nummer | Datum | Beschreibung | Zeit (Geplant) | Zeit (Effektiv) |
| 1.1 | 10.11 | Das Programm begrüsst den Benutzer und sagt was er/sie zu tun hat | 10 | 10 |
| 2.1 | 10.11 | Das Programm fragt den Benutzer nach seinem Namen | 20 | 5 |
| 3.1 | 10.11 | Der Benutzer kann seinen Vor- und Nachnamen eingeben | 10 | 5 |
| 4.1 | 10.11 | Das Programm erstellt aus den Namen von selbst eine E-Mailadresse | 10 | 10 |
| 5.1 | 10.11 | Alle Workshops werden zur Auswahl angezeigt | 20 | 10 |
| 6.1 | 10.11 | Das Programm fragt in welchen Workshop man sich einschreiben will und dass man als erstes sein liebster Workshop schreiben soll. | 10 | 5 |
| 7.1 | 10.11 | Der Benutzer kann seine drei Auswahlen in die Konsole schreiben | 10 | 5 |
| 8.1 | 10.11 | Das Programm merkt sich die drei Auswahlen des Benutzers | 10 | 10 |
| 9.1 | 10.11 | Das Programm fragt, ob der Benutzer noch weitere Personen anmelden will. | 10 | 5 |
| 10.1 | 10.11 | Der Benutzer kann weitere Namen angeben und drei Workshops pro Person wählen | 20 | 8 |
| 11.1 | 10.11 | Falsche Namen und Workshopnamen werden abgefangen | 20 | 10 |
| 12.1 | 10.11 | Verbesserungsvorschläge werden rot angezeigt | 30 | 10 |
| 13.1 | 10.11 | Es wird eine Liste aller Personen und deren gewählten Workshop ausgegeben | 10 | 5 |
| 14.1 | 10.11 | Das Programm fragt dann, ob die Workshopeinteilung starten soll. | 10 | 1 |
| 15.1 | 10.11 | Das Programm teil automatisch alle Personen in ihren liebsten Workshop ein | 50 | 20 |
| 16.1 | 10.11 | In einem Workshop hat es nur 4 – 20 Plätze frei! | 30 | 20 |
| 17.1 | 10.11 | Sobald ein Workshop voll ist, werden die nachfolgenden Personen in ihren zweit liebsten Workshop eingeteilt | 30 | 15 |

1. Kontrollieren

## **Testprotokoll**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| 1.1 | 10.11 | OK | Arpagaus |
| 2.1 | 10.11 | OK | Arpagaus |
| 3.1 | 10.11 | OK | Arpagaus |
| 4.1 | 10.11 | OK | Arpagaus |
| 5.1 | 10.11 | OK | Arpagaus |
| 6.1 | 10.11 | OK | Arpagaus |
| 7.1 | 10.11 | OK | Arpagaus |
| 8.1 | 10.11 | OK | Arpagaus |
| 9.1 | 10.11 | OK | Arpagaus |
| 10.1 | 10.11 | OK | Arpagaus |
| 11.1 | 10.11 | OK | Arpagaus |
| 12.1 | 10.11 | OK | Arpagaus |
| 13.1 | 10.11 | OK | Arpagaus |
| 14.1 | 10.11 | OK | Arpagaus |
| 15.1 | 10.11 | OK | Arpagaus |
| 16.1 | 10.11 | OK | Arpagaus |
| 17.1 | 10.11 | OK | Arpagaus |

Einige dieser Funktionen mussten angepasst werden um miteinander zu Funktionieren und manche brauchten z.b noch ein Try-Catch

1. Auswerten

Das gute ist das unsere Gruppe das Projekt frühzeitig fertig hatte und dadurch hatten wir mehr Zeit für das Debugging. Da die Projektdokumentation viel Zeit in Anspruch nahm, hatte 1 Gruppenmitglied weniger zu Programmieren.