

MR2

Management Review

"Taco" - QSE SEPM SS09 Gruppe 2

Projektbeschreibung

Ausgangssituation

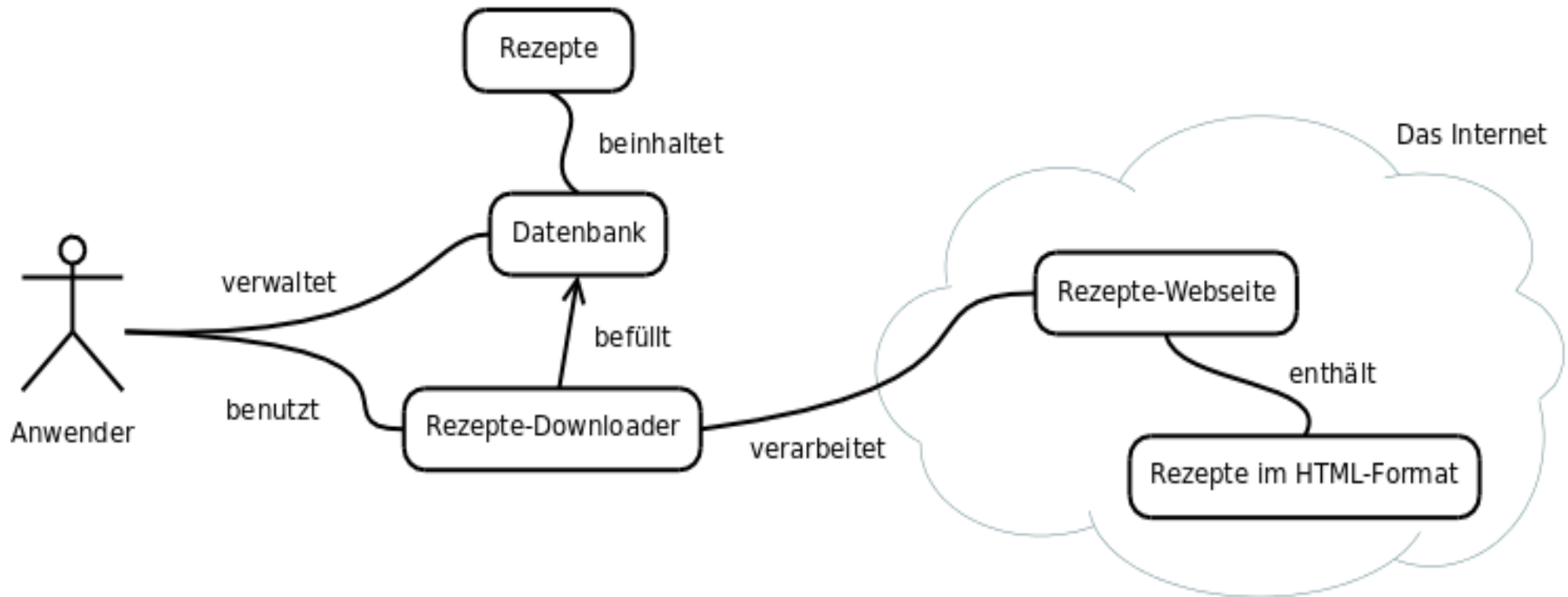
- Koch- Rezepte weit verteilt (Web, Kochbücher, etc.)
- Oft unübersichtlich, schlecht durchsuchbar

Ziele

- Übersichtliche, gut durchsuchbare Datenbank
- Software für ernährungsbewusste Personen
- Rezepte an Hand von Schlagwörtern suchen
- Berücksichtigung des „virtuellen Kühlschranks“
- Rezepte selber gestalten

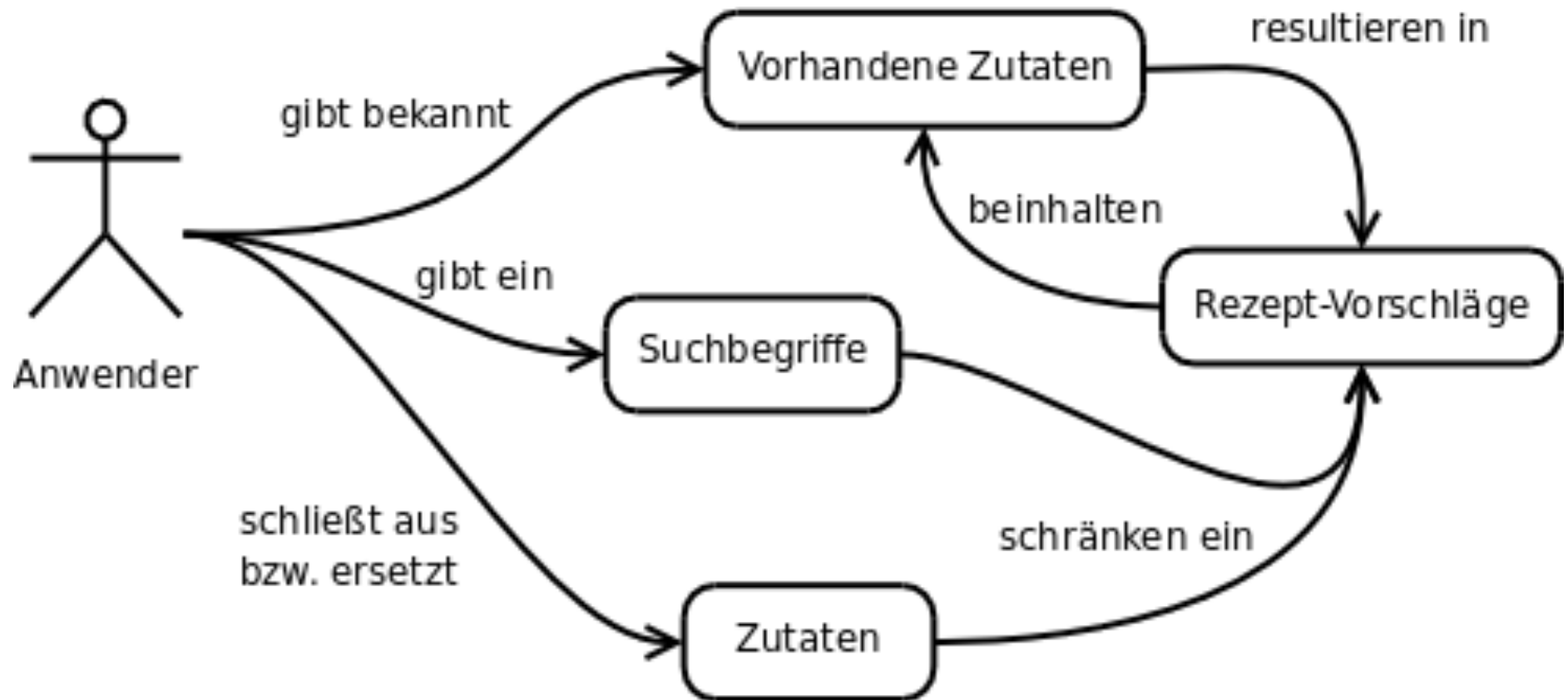
Funktionalität

- Rezeptverwaltung



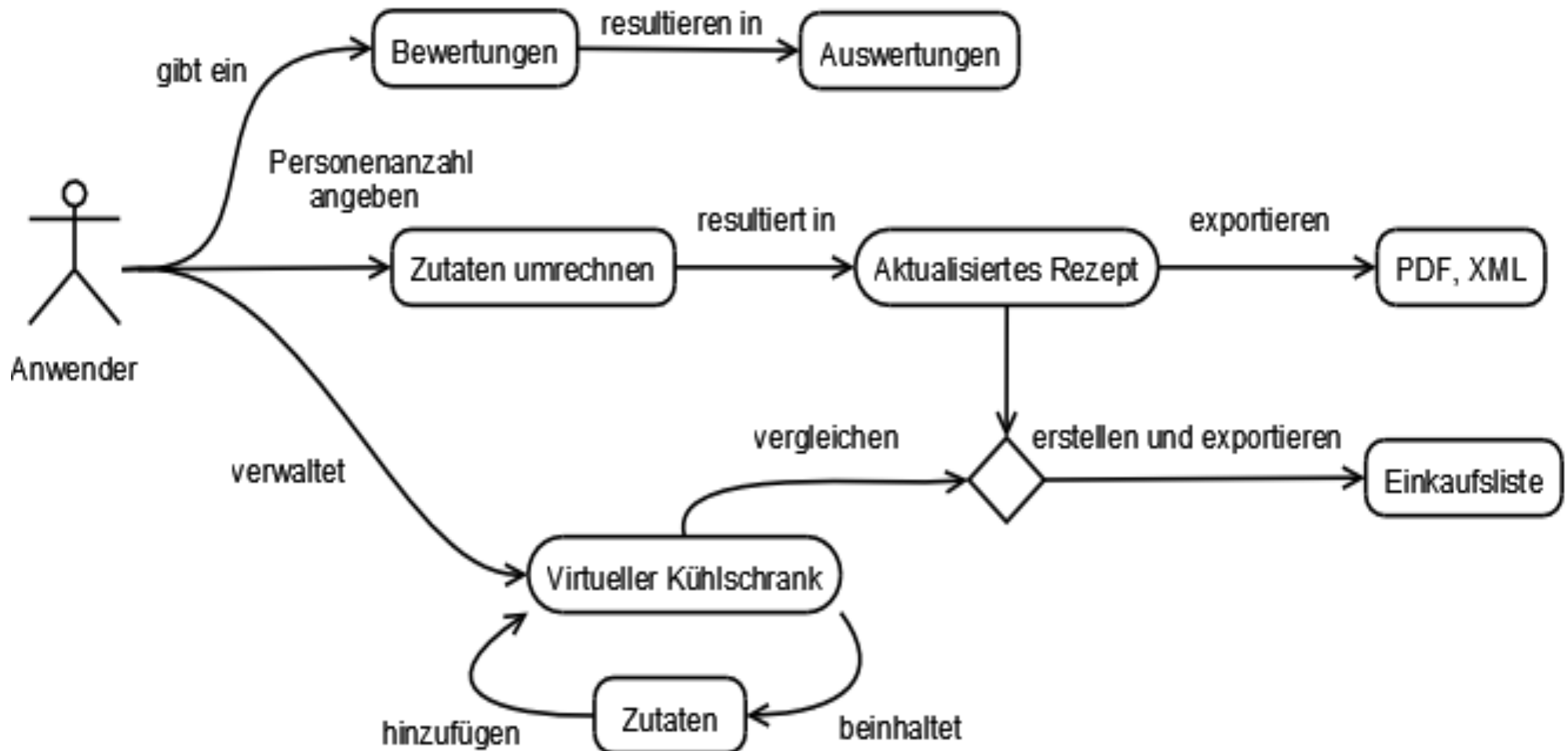
Funktionalität

- Rezeptsuche



Funktionalität

- Verarbeitung der Rezepte



Pflicht-Features

1. Manuelles Eingeben und Korrigieren von Rezepten
2. Erstellen von Rezeptvorschlägen nach Eingabe von vorhandenen Zutaten
3. Rezepte von Websites importieren
4. Verwaltung ähnlicher Zutaten
5. Einbeziehen ähnlicher Zutaten bei der Suche
6. Ausschließen und Ersetzen von bestimmten Zutaten
7. Rezepte als ausdrucksbares Dokument exportieren
8. Bewertung von Rezepten durch Benutzer
9. Statistische Auswertungen (am meisten gekocht, am besten bewertet, etc.)

Optionale Features

1. Anzeigen von Bildern zu den Rezepten
2. Virtueller Kühlschrank (merken von vorhandenen Zutaten zwischen Programmaufrufen)
3. Liste der fehlenden Zutaten als Einkaufsliste anzeigen und exportieren
4. Themen-Suche (z.B. asiatisch)
5. Zutaten-Menge auf eine bestimmte Personenanzahl umrechnen
6. Kalorienberechnung der Speisen + Suchkriterium Kaloriengehalt
7. Interface für Barcode-Scanner (Aktualisieren des virtuellen Kühlschranks)

Timeline

- 20.04 - Vorbereitung
- 28.04 - MR1 / JourFixe2009-04-23
- 01.05 - Projektauftrag / JourFixe2009-04-30
- 07.05 - PaperTaco
- 18.05 - Taco 0.1
- **20.05 - MR2 / Taco 0.2**
- 28.05 - Taco 0.4
- 04.06 - Taco 0.6
- 11.06 - Taco 0.8
- 18.06 - Taco 0.9
- 25.06 - Taco 1.0 RC
- 28.06 - Taco 1.0 Release
- ??..06 - MR3



paper taco
30. april ~ 7. mai

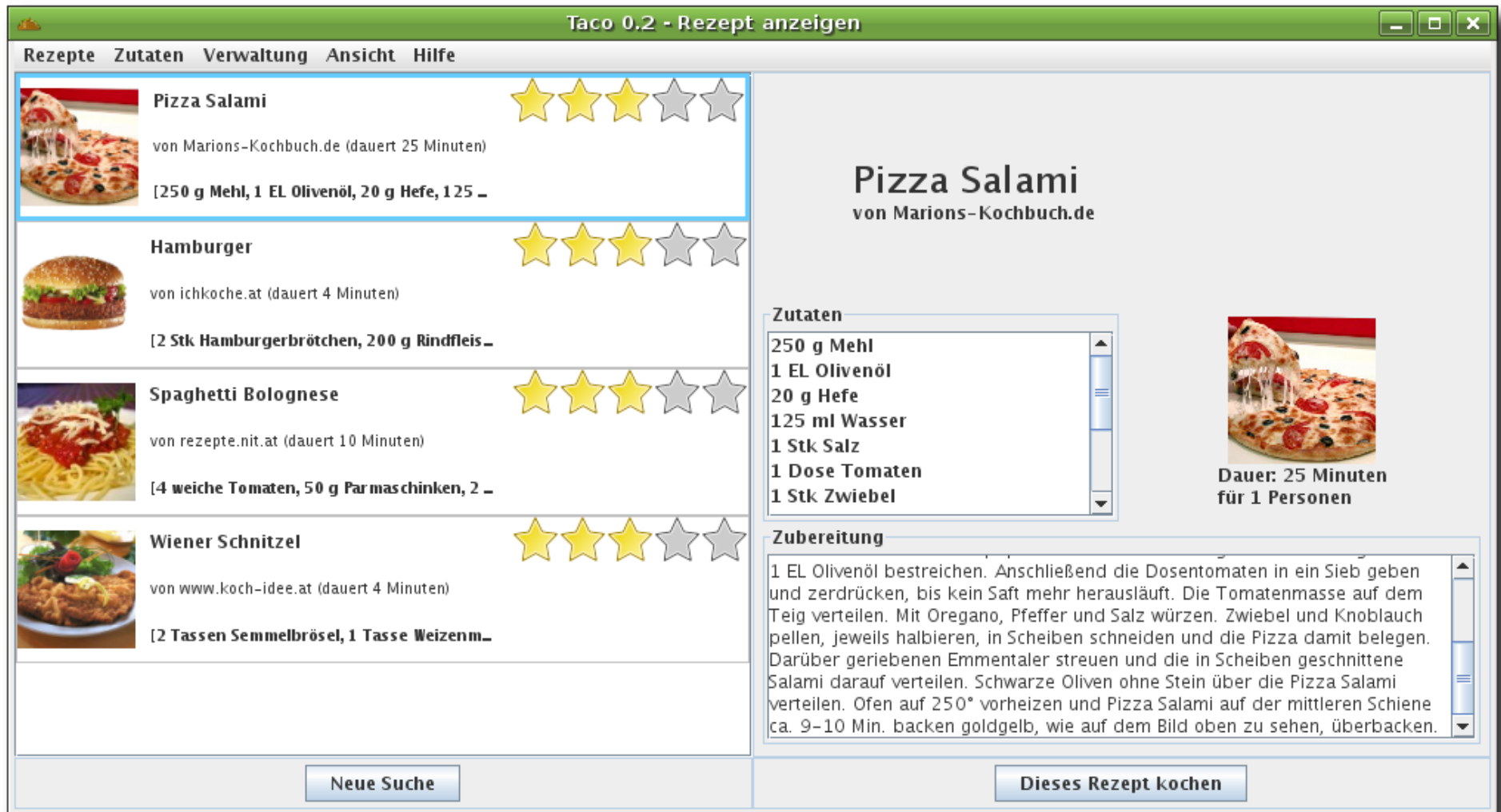


Taco 1.0
28. juni 2009

Statusbericht

- Fertigstellung und Korrektur des Projektauftrags
- Papier-Prototypen für GUI-Design
- Dokumentation: Komponentendiagramm, DB-Beschreibung, Klassendiagramm, UI-Beschreibung, Testplan, Testdokument, WBS und GANTT-Diagramm, Deployment-Diagramm
- Code: Model-Klassen, DAOs, GUI, Controller, Unit-Tests
- GUI: Guter Fortschritt, angepasste Komponenten (Demo)
- Web-Import: Jython-Integration funktioniert schon (Demo)

Prototyp



Next Steps

- Lauffähiger Prototyp mit Features 1-3
- Behebung der aufgetretenen Fehler im Programm
- Erweitern der Testfälle
- Implementierung der weiteren Features