

Protokoll Milestone 2

Datum: 19.05.2021

Beginn: 14:41

Ende: 15:03

Anwesende: Prof. Dr. Dirk Hesse

Abwesend: 0

Alemi, Ranna

Entschuldigt: 0

Brecht, Tim

Dobler, Anna Marie

Schuler, Sven

Schwind, Timo

Topic 1: Vorgehen

Darlegung des bisherigen Vorgehens und der Arbeitsschritte mit Hilfe des Gantt Diagramms.

Erläuterung des Vorgehens bei der ERM Erstellung.

Topic 2: ERM 1- Use Case 1

Vorstellung des ERM für Use Case 1 (Hochladen des CSV-files)

Hierbei werden die Daten in der Tabelle „raw data“ gespeichert und zur Weiterverarbeitung in die „account movement“ Tabelle gespiegelt.

Über die „transactions“ Tabelle werden beispielsweise die gezahlten und zu zahlenden Mietkosten den Mietern zugeordnet, ebenso sollen hier die Transaktionen den Mietern zugeordnet werden.

Der Mieter wird dem entsprechenden Mietvertrag („lease contract“) zugeordnet. Mieten werden in die „transactions“ Tabelle eingepflegt.

Mit Hilfe eines Vorzeichens sollen die Ein- und Auszahlungen in der „transaction“ Tabelle unterschieden werden. Einzahlung positives, Auszahlung negatives Vorzeichen. Dies soll zur Erkennung von Rückzahlungen dienen.

Informationen über den Mieter werden in den Stammdaten gehalten.

Anmerkungen/Verbesserungsvorschlag Herr Hesse

Überdenken, ob es besser wäre von der „transaction“ Tabelle direkt zu dem Mietvertrag zu gehen, anstelle über die „renter“ Tabelle.

Dies würde zum Einsatz kommen, wenn Kosten aufkommen, die das gesamte Haus betreffen.

Sonst keine weiteren Einwände.

Topic 3: ERM 2- Use Case 2

Vorgehen verhält sich analog zu ERM 1.

Anmerkungen/Verbesserungsvorschlag Herr Hesse

Überdenken, ob die Entity „renter“ relevant ist. Die Eigenschaften des Mieters, wie Name sind hier nicht relevant. Die Anzahl der Mieter oder die Quadratmeterzahl der Wohnung ist hier vielmehr ausschlaggebend.

Topic 4: ERM 3- Use Case 3

Vorgehen verhält sich analog zu ERM 1.

Anmerkungen/Verbesserungsvorschlag Herr Hesse

Zur Darstellung der Zahlungsunterschiede Soll und Haben miteinführen.

Anmerkung Teammitglied: Eventuell wie bei Transaktionstabelle über Vorzeichen lösbar.

Topic 5: Weitere Schritte

In Milestone 3 soll ein logisches Datenmodell vorliegen. In diesem Schritt wird dann auch klar, ob Entitäten oder ähnliches vergessen wurden. Außerdem wird hier die Art der Datenbank ausgewählt (z.B. MongoDB; relationale DB) die Wahl des Herstellers spielt noch keine Rolle. In diesem Fall soll eine relationale Datenbank erstellt werden. Das Ziel besteht darin mit Hilfe von SQL DBM zweidimensionale Modelle in Relationenmodellen, einer Datenbank abzubilden. Außerdem sollen Skripte für den SQL Server erstellt werden.

Das dabei entstandenen logische Datenmodell wird im Milestone 3 besprochen. Ebenso könnte sich im Laufe des Prozesses die Frage auftun, ob eventuell eine andere Reihenfolge der ERM Diagramme benötigt wird, um die Anzahl der Joins zu verkleinern.

Die Aufgaben bis zum nächsten Milestone bestehen in der Erstellung des logischen Datenmodells, der Überprüfung, dass keine Normalform verletzt wird sowie die Erstellung des casetools.

Es wird empfohlen bereits in dieser Phase mit der Programmierung des Gerüsts zu beginnen, das beinhaltet beispielsweise die Connections und die Stammdatenpflege.

Topic 6: Fragen

Auf die Frage, ob auch auf der Oberfläche Stammdaten hinzuzufügen sind oder ob diese im Hintergrund in der Datenbank hinzugefügt werden, wurde sinngemäß folgendes genannt:

Es sollte beides umgesetzt werden. Für ein Stammdatum soll ein Mietvertrag mit einem Klick generiert werden. Dies ist aber nicht für alle Stammdaten nötig. Es soll ein Skript geschrieben werden, das man einfügen und laufen lassen kann.

Die Vergabe von Rechten ist nicht gefordert. Es soll aber eine Auswahlmaske geben, mit welcher beispielsweise die Daten aus der CSV-Datei importiert und mit welcher Offene Posten angezeigt werden können.

Topic 7: Ende

Die Besprechung wurde um 15:03 beendet.

Protokoll Milestone 3

Datum: 02.06.2021

Anwesende:	Prof. Dr. Dirk Hesse	Abwesend:	0
	Alemi, Ranna	Entschuldigt:	0
	Brecht, Tim		
	Dobler, Anna Marie		
	Schuler, Sven		
	Schwind, Timo		

Topic 1: Vorgehen

Darlegung des bisherigen Vorgehens und der Arbeitsschritte mit Hilfe des Gantt Diagramms.

Erläuterung des Vorgehens bei der ERM Erstellung.

Topic 2: Inhalt

In diesem Milestone soll nicht genauer auf die Besprechung des Frontends und Backends eingegangen werden. In diesem Milestone ist die Erstellung eines funktionsfähigen Datenmodells relevant.

Topic 3: Entitäten

Da sich die Aufteilungen der Entitäten und deren Beziehungen während der Bearbeitung geändert haben, wurde die neue Aufteilung dargestellt und besprochen.

Es wurde aufgezeigt, dass die Tabellen in der BCNF vorliegen

Anmerkung/Erläuterungen zu den Entitäten/Beziehungen:

Die Entität Raw Data dient immer noch zur Dateneinpfege. Hier sind alle Attribute gemeinsam der Primary Key. Dadurch soll ein doppeltes Hochladen der csv-Datei – auch in Teilen- verhindert werden. Über einen Trigger werden die Attribute in Account Movement übernommen.

Die Entität Account Movement enthält einen Primary Key, der sich aus dem Wert der Attribute ergibt, die übernommen werden.

Eine Account Movement kann auf mehrere Transactions verteilt werden, wenn diese zu mehreren Mietern gehört. Gibt es zum Beispiel eine Account Movement für die Wasserabrechnung, die zu allen Mietern in einem Haus gehört, so wird diese auf mehrere Transactions anteilig verteilt. Für jeden Mieter liegt dann der Anteil, den er bezahlen muss in der Transaction Entität vor.

Die Tabelle Transactions weist Teilschlüsselabhängigkeiten auf. Diese wurden bewusst so gewählt, da PostingDate und Payment Reason für die Betriebskostenabrechnung nötig sind und durch diese Attribute kein JOIN angewendet werden muss. Herr Hesse weist darauf hin, dass durch das Datum eine Eindeutigkeit hergestellt werden kann, da zum Beispiel ein Auto nur an einem Ort zu einem Zeitpunkt sein kann.

Bei der Entität Lease Contract weist die Tabelle transitive Abhängigkeiten auf, ApartmentID lässt auf Security Deposit schließen. Deswegen wurde Security Deposit in Apartment Tabelle verschoben. Unter den zwei Annahmen (siehe PPP) gibt es weitere Schlüssel Kandidaten.

Die Entität Apartment weist transitive Abhängigkeiten auf, deswegen wurde eine extra Tabelle für die Häuser angelegt, da alle Apartments im gleichen Haus die gleiche Adresse haben. Hinweis: Housenumber ist ein künstlicher Schlüssel und nicht die Hausnummer der Anschrift. Um Redundanzen zu eliminieren, wurde die PLZ und Stadt in eine weitere Tabelle ausgegliedert.

Die Anschrift wurde in das Apartment übernommen, sodass die Lokalisierung des Apartments möglich ist, auch wenn das Apartment nicht bewohnt ist. Das ist möglich, da die Anschrift des Mieters über das Apartment, in dem er wohnt, dem Mieter zugeordnet werden kann.

Topic 4: Anmerkung von/ Diskussion mit Herrn Hesse:

Die Aufteilung der Transaktion Entity ist gut, da hier verschiedene Sachen/Vorgänge betroffen sein können.

Der Import der csv Datei und der Betriebskostenabrechnung ist über die vorliegende Logik möglich.

Eine Anmerkung bezog sich auf die Offenen Posten. Hier wurde gefragt, wo man die Gegenposition herbekommt.

Über den FK in der Transaction Entität kann der zugehörige Mieter bestimmt werden. Eine Stored Procedure, die automatisch monatlich ausgeführt wird, werden alle aktiven Mietverträge in die Transaction Entität aufgenommen.

Die Zuordnung der Offenen Posten und der Betriebskosten soll über den Verwendungszweck erfolgen.

Zu beachten ist, dass sich die Soll Positionen der Miete über die Jahre ändern kann, da sich die Miete im Mietvertrag geändert werden könnte. Deswegen wurde das Attribut „active“ in die Entität Lease Contract mit aufgenommen, anhand derer man erkennen kann, welche Verträge noch gültig sind und monatlich in die Transactions als Minusposition (Soll) mitaufgenommen werden müssen.

Anschließend kann nach der RenterID kumuliert werden und so festgestellt werden, ob der Mieter nachzahlen muss, etwas zurückbekommt oder ob alle Zahlungen richtig erfolgt sind.

Transaction enthält die runtergebrochenen Zahlen der csv-Datei. Eine darauf gelegte View zeigt die Höhe der Zahlungen eines Mieters.

Bei der entity Lease contract sind die historischen Daten wichtig. Dadurch ist es möglich eine Historie der Mietverträge für jeden Mieter zu erstellen.

Topic 5: Inhalt Dokumentation

Vorstellung des Frontend Mockups mit Hilfe von flickr

Topic :6 Inhalt Dokumentation

Inhalt des Projekts, Anforderungen, Realisation, Frameworks, Lessons learned (auch in Form von Schwierigkeiten), Modelle als PowerPoint

Die Dokumentation ist ebenso auf GitLab einzustellen.

Topic 7: Fragen

Die automatische Generierung der Skripte sind eigentlich eine Anforderung, welche aber nicht mehr zwangsläufig eingearbeitet werden müssen.

In GitLab soll ein SQL Skript hochgeladen werden, das alle nötigen SQL Statements enthält, um die Datenbank aufzusetzen.

Die Testdaten können über den Insert Befehl eingefügt werden.

Zur Veranschaulichung der Stammdatenpflege soll ein Beispiel angelegt werden.

Topic 7: Ende

Der Milestone wurde beendet.

Protokoll Milestone 4

Datum: 16.06.2021

Anwesende:	Prof. Dr. Dirk Hesse	Abwesend:	0
	Alemi, Ranna	Entschuldigt:	0
	Brecht, Tim		
	Dobler, Anna Marie		
	Schuler, Sven		
	Schwind, Timo		

Topic 1: Vorgehen

Vorstellung des Gantt-Diagramms.

Topic 2: Aufbau und Vorstellung Anwendung

Die Anwendung basiert auf einem SQL Server und einer Springboot Anwendung mit Rest Api (als Backend). Das Frontend wird mit einer Angular App realisiert.

Die Startseite enthält ein Menu und einen Homebutton.

Stammdaten

Die Stammdaten zu Renter, Apartment, etc. werden in der Übersicht der einzelnen Stammdaten gezeigt. Als Id wurde eine fünfstellige Zahl mit 10 Inkrement verwendet. Dadurch kann über die letzte Ziffer zwischen den einzelnen Arten der Stammdaten unterschieden werden. Renter zum Beispiel enden auf 0; LeaseContract enden auf 2.

In der Übersicht ist jede Zeile anklickbar, dadurch kommt man auf die Details. Hier ist die Id, sowie weitere fixe Eingabefenster ausgegraut und in HTML als disabled gesetzt. Nicht ausgegraute Zellen können bearbeitet werden. Diese Änderungen gelangen über den safe button in die Datenbank.

Über den new button können neue Stammdaten angelegt werden.

Anmerkung: Delete button muss bei renter noch hinzugefügt werden.

Zusätzlich steht eine Suchfunktion als extra feature zur Verbesserung der Usability zur Verfügung.

Die Adresse der Apartments wird mit Hilfe eines Joins über die Apartmentnummer und Häuser zugeordnet. Die Stadt wird über eine in der Datenbank verankerten Tabelle über die Postleitzahl hinzugefügt.

Transaktionale Funktionale Fähigkeiten

Open Position

Die entsprechenden Datenbank Einträge werden für die einzelnen Mieter aufsummiert und dadurch erhält man die Balance. Positive Einträge bedeuten, dass der Mieter zu viel gezahlt hat.

Zwischenfrage von Herrn Hesse: Wie ist die Übersicht/ Open Positions aktualisierbar?

Hier werden über eine Stored Procedure bei jedem Neuladen die Daten aktualisiert. Auch wenn es besonders bei einer großen Anzahl an gleichzeitigen Aufrufen zu Performance Problemen kommen kann, ist hier die Aktualität der Daten von höherer Relevanz für uns.

Operating Cost Statement

Hier wird nach der RenterId und Jahr gruppiert. Aktive zeigt an, ob der Mieter einen laufenden Mietvertrag hat.

Über ein Get statement erhält man die Excel Tabelle. Hier wird die Anschrift reingeladen. Das Operating Cost Statement ähnelt einem jährlichen Kontoauszug.

Frage von Herrn Hesse: Ist das (Positionen des Operating Cost Statements) kumuliert? Ist das bereits die Betriebskostenabrechnung?

Aktuell liegt noch keine gesetzlich zulässige Betriebskostenabrechnung vor, vielmehr eine Art Kontoauszug. Hierbei kommen alle Daten aus der Transaktionstabelle, die sich auf einen Mieter beziehen.

Anmerkung Herr Hesse:

Es soll eine Betriebskostenzusammensetzung erstellt werden, die ein paar Einzelpositionen aufzeigt und nicht nur eine Zahl der aggregierten Betriebskosten.

Hier wurde der Renter 10010 vorgezeigt, dieser besitzt bereits eine Unterteilung in Strom und Wasser.

Anmerkung Herr Hesse:

Hier soll noch ein Teiler mit reingebracht werden. Es soll sowohl die Gesamtheit/ Salden der Betriebskosten als auch einen Schlüssel dazu ausgegeben werden, sowie Einzelbetriebskosten.

Dies ist noch zu beheben.

Zur Erstellung dieser Betriebskostenabrechnung kann die eigene Vorlage/ Excel Template benutzt werden.

Statement of bank account

Zeigt einen Auszug aller Daten, die hochgeladen wurden, wie beispielsweise in Form von Überweisungen, Daueraufträgen. Über eine Stored Procedure werden die relevanten Daten über den Betreff, Begünstigter etc. erfasst und auf einzelne Transaktionen verteilt.

CSV-File upload

Hier wurde ein CSV-file, dass es nicht möglich ist, ein CSV-File doppelt hochzuladen. Die hochgeladenen Daten laden zunächst unverändert in einer Raw Data Tabelle. Ein Trigger wartet auf insert statements und kopiert dann die Daten in die Account Movement Tabelle.

Topic 3: Anmerkung von Herrn Hesse:

Wenn die Gruppe möchte, darf dieses Projekt präsentiert werden. Hierbei sollten auch die verwendeten stored procedures, trigger und ähnliches vorgestellt und erklärt werden. Die Präsentation ist auf Englisch und wird, sofern von den Präsentierenden erlaubt, aufgenommen. Über den Wunsch der Gruppe, ob präsentiert wird und ob eine Aufnahme möglich ist, wird Herr Hesse nach Besprechung in der Gruppe per Mail informiert.

Auf Grund des weiten Fortschrittes wird der nächste Milestone entfallen.

Anmerkung Herr Hesse: Das Projekt gefällt ihm sehr gut und es besteht die Möglichkeit zur Präsentation sowie die Aussicht auf Extrapunkte.

Topic 4: Inhalt Dokumentation

Die Abgabe wird nach Korrektur der angesprochenen Änderungswünsche über das gitlab der Hochschule Herrn Hesse zur Verfügung gestellt. Dort wird auch die Dokumentation und alle weiteren Dokumente vorliegen.

Anmerkung Herr Hesse:

Er benötigt nicht nur Gastrechte, um auf das Projekt in gewünschter Weise zugreifen zu können.

Topic 5: Ende

Der Milestone wurde beendet.