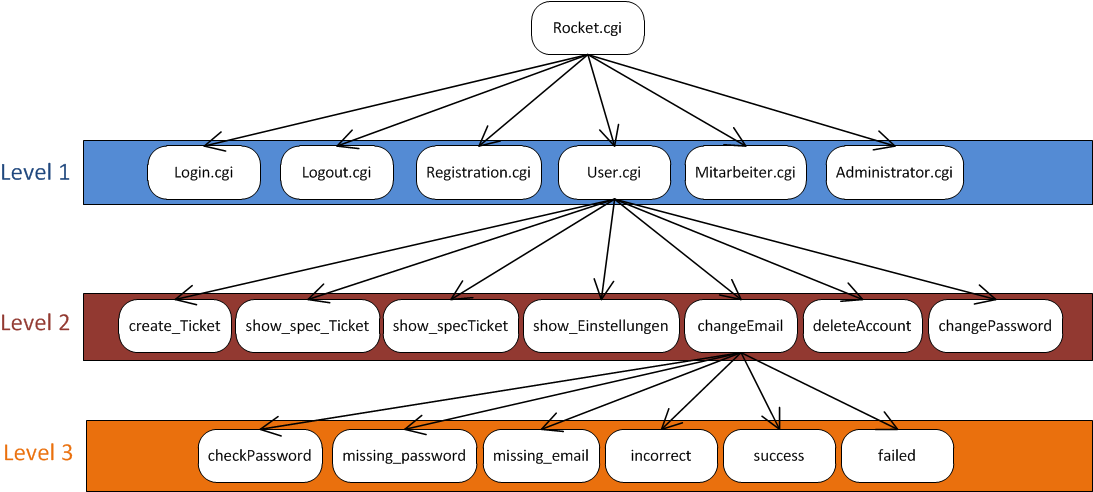
**Programmstruktur:**

Am Anfang war geplant, dass ein einzelnes Skript den gesamten Ablauf bzw den dargestellten Websiteninhalt steuert und der Inhalt in Subroutinen ausgelagert wird.

Dies erwies sich nach kurzer Zeit aber als problematisch, da es in unserem Konzept 3 Benutzergruppen mit verschiedenen Rechten gibt (User, Mitarbeiter, Administrator) und so das einzelne „Steuer-Skript“ rasch sehr umfangreich wurde.

Auch die dauerhafte gleichzeitige Bearbeitung des Skripts durch 2 Entwickler erwies sich trotz Versionsverwaltung als fehleranfällig und wir entschlossen uns, die Projektstruktur zu redesignen und in Teilbereiche aufgeteilt.

Die Steuerung des Ablaufs übernimmt nun nicht mehr ein Skript sondern alle Skripte, die in einer Art Baumstruktur angeordnet sind.

Es gibt 3 Variablen, die zur Steuerung des Ablaufs genutzt werden. Diese sind in der Session verfügbar und lauten Level1, Level2, Level3.

Für dieses Beispiel wurde die „Passwort ändern“-Funktion des Users herausgenommen.

In Level1 steht „User“, deshalb leitet die Rocket.cgi den Aufruf an User.cgi weiter.

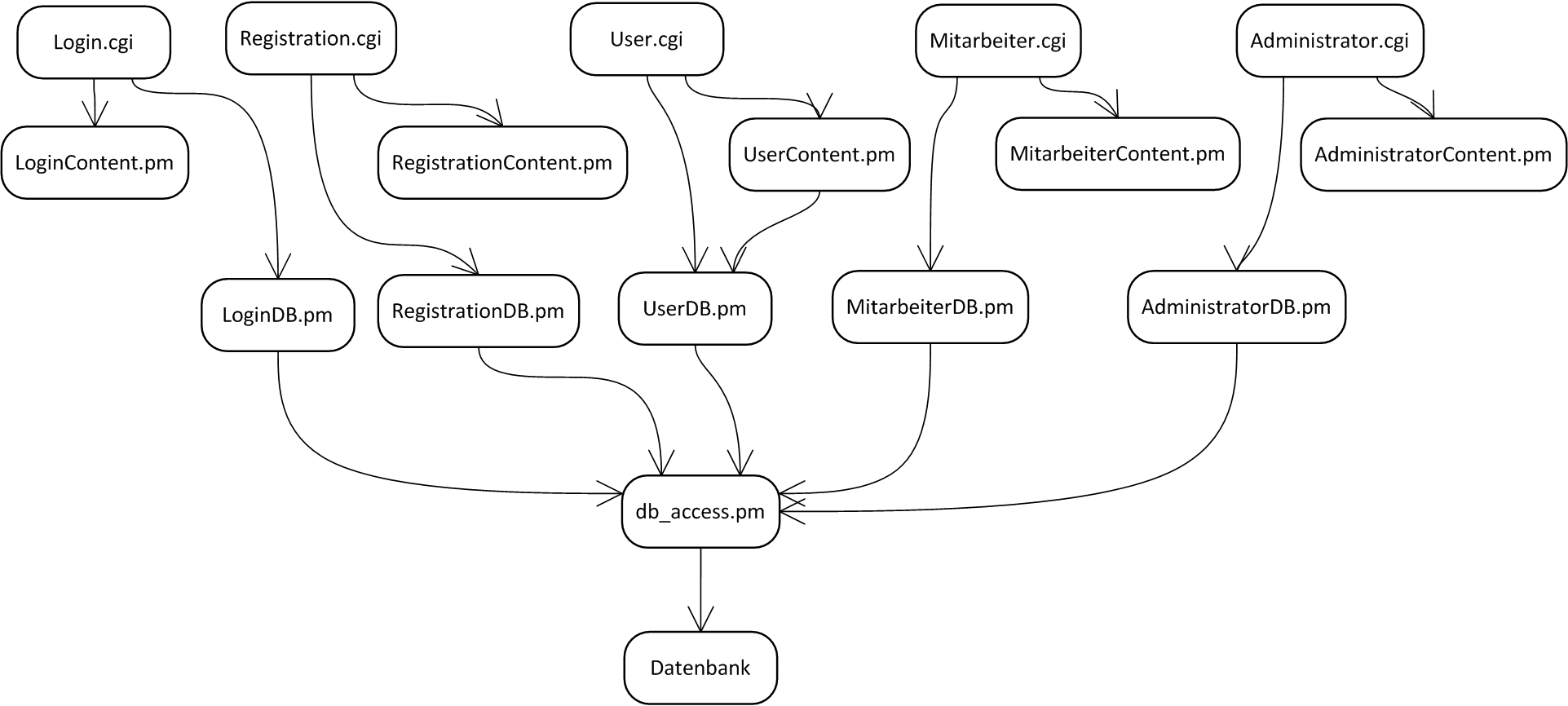
User.cgi entscheidet nach dem Level 2-Wert wie nun weiter gemacht wird. In diesem Fall lautet der Wert „changeEmail“ und je nachdem auf was Level 3 gesetzt wurde, wird das eingegebene Passwort überprüft („checkPassword“), Fehlermeldungen ausgegeben (Passwort fehlt, Email-Adresse nicht angegeben, …) oder eine Erfolgsmeldung ausgegeben.

Jedesmal, wenn der Benutzer eine Eingabe macht, (Formulare ausfüllen/Links anklicken) werden die entsprechenden Level1/2/3-Values gesetzt und wieder Rocket.cgi aufgerufen. Nun startet der Auswahlprozess von neuem.

Die „Level1-Skripte“ (User.cgi, Mitarbeiter.cgi, Administrator.cgi, Registration.cgi , Login.cgi) verfügen über Module, welche für die Websitendarstellung (Content) zuständig sind und für die Datenbankanbindung (Datenaustausch).

Die Namensgebung wurde deshalb auf -Skriptname-DB.pm und –Skriptname-Content.pm festgelegt um Übersichtlichkeit zu gewährleisten.

Allerdings darf nur ein Modul (db\_access.pm) direkt mit der Datenbank kommunizieren um die Wartbarkeit zu optimieren, falls im späteren Entwicklungsverlauf auf ein anderes Datenbanksystem zurückgegriffen werden muss.

**Vereinfachte Darstellung:**

Das Skript „SaveDataForm.cgi“ spielt auch eine zentrale Rolle, indem es die Benutzereingaben entgegennimmt und diese in der zugehörigen Session speichert. Danach wird wieder zum Einstiegspunkt (Rocket.cgi) weiterleitet.

Dies ist unerlässlich, da bei unseren HTML-Formularen die Daten per POST-Methode übermittelt werden. Dabei muss ein Skript angegeben werden, dass diese Daten entgegennimmt und weiterverarbeitet. Um Redundanz zu vermeiden verwenden wir uns hierfür entschieden ein Skript einzusetzen, welches universell einsetzbar ist „SaveDataForm.cgi“.