Alexandru Carp
30433

Patient ECG History Database
Use-Case Model

Version <1.0>

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

Use-Case Model

1.Use-Cases Identification

Use case: Manage patient ECG Data
Level: summary level
Primary actor: physician
Main success scenario: log in, CRUD on data, annotate, log out
Extensions: <>

Use case: Review patient ECG Data
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, read data, log out
Extensions: fail to log in, fail to find data

Use case: Add ECG recording
Level: user-goal level
Primary actor: physician
Main success scenario: log in, perform ECG, add data, annotate, log out
Extensions: fail to validate date, fail to connect to server

Use case: Update ECG recording
Level: user-goal level
Primary actor: physician
Main success scenario: log in, select ECG, annotate, log out
Extensions: fail to find ECG, fail to validate changes, fail to connect to server

Use case: Review patient ECG Data
Level: user-goal level
Primary actor: physician
Main success scenario: log in, read ECG, log out
Extensions: fail to find ECG

Use case: Delete patient ECG Data
Level: user-goal level
Primary actor: physician
Main success scenario: log in, select ECG, delete ECG, log out
Extensions: fail to connect to server

Use case: Add patient
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, add new patient, log out
Extensions: fail to validate data, fail to connect to server

Use case: Add physician
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, add new physician, log out
Extensions: fail to validate data, fail to connect to server

Use case: Review physician
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, review physician data, log out
Extensions: fail to find physician

Use case: Review patient
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, review patient data, log out
Extensions: fail to find patient

Use case: Update physician
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, update physician, log out
Extensions: fail to find physician, fail to connect to server

Use case: Update patient
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, update patient, log out
Extensions: fail to find patient, fail to connect to server

Use case: Remove physician
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, remove physician, log out
Extensions: fail to find physician, fail to connect to server

Use case: Remove patient
Level: user-goal level
Primary actor: insurer
Main success scenario: log in, remove patient, log out
Extensions: fail to find patient, fail to connect to server

Use case: Review personal data
Level: user-goal level
Primary actor: patient
Main success scenario: request access, enter credentials, access requested data
Extensions: fail to find user

Use case: Log in
Level: sub-function level
Primary actor: physician
Main success scenario: enter credentials, access physician interface
Extensions: fail to match credentials, fail to connect to server

Use case: Log out
Level: sub-function level
Primary actor: insurer
Main success scenario: enter credentials, access insurer interface
Extensions: <>

Use case: Log out
Level: sub-function level
Primary actor: physician
Main success scenario: log out
Extensions: <>

Use case: Search for user
Level: sub-function level
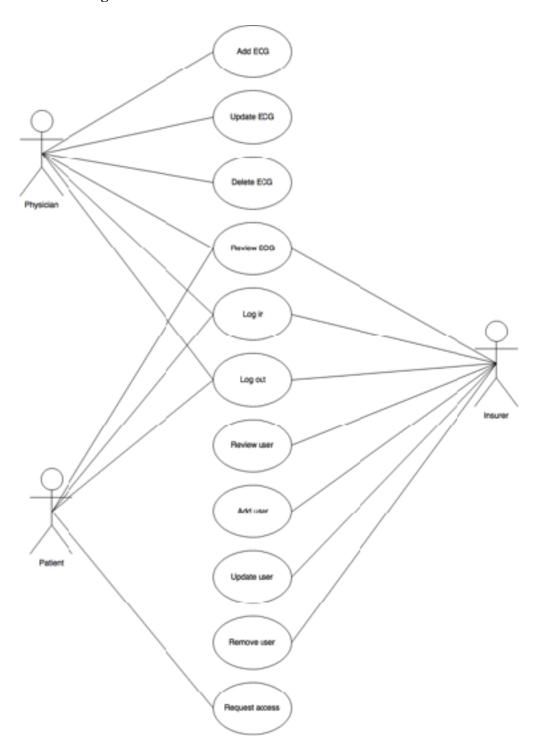Primary actor: physician/insurer
Main success scenario: search for a name in the system and return the result
Extensions: display a message if name is non-existent

2.UML Use-Case Diagrams