- Systemanalyse -

QuantumCryptoCram

Version: 1.8

Projektbezeichnung	QuantenCryptoCram	
Projektleiter	Johannes Sporrer	
Verantwortlich	Johannes Sporrer	
Erstellt am	02.04.2021	
Zuletzt geändert	22.06.2021 0:45	
Bearbeitungszustand		in Bearbeitung
	X	vorgelegt
		fertig gestellt
Dokumentablage	git.oth-aw.de/swp/sose2021/team_c/quantumcryptocram/-/tree/develop/01_Analyse	

Änderungsverzeichnis

Änderung			Geänderte Kapitel	Beschreibung der Änderung	Autor	Zustand
Nr.	Datum	Version				
1	02.04.21	0.0A	Alle	Initiale Produkterstellung	Benedikt Bartl	Fertig
2	09.04.21	0.0B	2, 3, 4	Initiale Überarbeitung der Kapitel	Johannes Sporrer	Fertig
3	15.04.21	0.0C	3	Use Cases aktualisieren	Johannes Sporrer	Fertig
4	17.04.21	0.1A	3	Use Cases verbessert	Johannes Sporrer	Fertig
5	19.04.21	0.1B	4	Nicht Funktionale Anforderungen	Johannes Sporrer	Fertig
6	22.04.21	0.2A	3	Use Cases verbessert	Johannes Sporrer	Fertig
7	23.04.21	0.3	3, 2	Verständlichkeit, Rechtschreibung	Christoph Kennerknecht	Fertig
8	27.04.21	0.4	3.2	UseCases finalisiert und formatiert	Felix Paris	Fertig
9	01.05.21	0.4	9	Glosar erstellt	Johannes Sporrer	Fertig
10	03.05.21	0.4	Alle	(Nicht mehr nötige) blaue Texte entfernt, Formatting	Florian Hofmann	Fertig
11	05.05.21	0.5	2, 3, 7	PDF Verweise eingefügt, Namenerklärung eingefügt, Abkürzungsverezeichniss begonnen	Johannes Sporrer	Fertig
12	06.05.21	0.6	Alle	Korrekturen	Johannes Sporrer	Fertig
13	06.05.21	1.0	Alle	Formatierung	Felix Paris	Fertig
14	14.05.21	1.1	3.2	UC16-2/3/4 + UC17-2a Kleinere Korrekturen	Johannes Sporrer	Fertig
15	17.05.21	1.2	3.2	UC 18-5 Kleine Ergänzung	Johannes Sporrer	Fertig

16	27.05.21	1.3	3.2	UC2 Korregiert UC 28 Name korregiert	Johannes Sporrer	Fertig
17	28.05.21	1.4	3.2	UC 12 Ablaufvariante 2b ergänzt	Sporrer, Kennerknecht, Paris	Fertig
18	17.06.21	1.5	3.2	UC-2, UC-4, UC-5, UC-15, UC-27 überarbeitet, UC-28 entfernt	Sporrer	Fertig
19	21.06.21	1.6	3.2	UC-7, UC-14, UC-21, UC24 bearbeitet, UC 28 eingefügt	Sporrer	Fertig
20	21.06.21	1.7	3.2	UC-17, UC26 bearbeitet	Sporrer	Fertig
21	21.06.21	1.8	3.2	UC-13, UC-14, UC-17, UC-21, UC-27	Sporrer	Fertig
22	22.06.21	1.9	3.2	UC-27	Kennerknecht, Bartl	Fertig

Prüfverzeichnis

Die folgende Tabelle zeigt einen Überblick über alle Prüfungen – sowohl Eigenprüfungen wie auch Prüfungen durch eigenständige Qualitätssicherung – des vorliegenden Dokumentes.

Datum	Geprüfte Version	Anmerkungen	Prüfer	Neuer Produktzustand
21.04.21	0.1B	Siehe "./ Pruefprotokoll_ Systemanalyse_v0-1b_Kennerknecht.docx"	Christoph Kennerknecht	Für V. 0.2
21.04.21	0.1B	Review Kapitel 3.2	Felix Paris	Für V. 0.2
22.04.21 - 23.04.21	0.2A	Review Kapitel 3.2	Jan Friedrich Florian Hofman Jakob Gotz Christopf Kennerknecht	Für V. 0.3
23.04.21	0.2	Review Kapitel 3.2	Zell Elias	0.4
23.04.21	0.3	Review Kapitel 3.2 u 3.4	Bartl Benedikt	0.4

Inhalt

1 Einleitung6
2 Ausgangssituation und Zielsetzung6
3 Funktionale Anforderungen7
3.1 Use-Case Übersicht7
3.2 Use-Case Beschreibungen8
3.3 (Sonstige) Funktionalität27
3.4 Modell des Problembereichs (Konzeptionelles Datenmodell)27
4 Nicht-Funktionale Anforderungen29
4.1 Benutzbarkeit (Usability)29
4.2 Zuverlässigkeit (Reliability)29
4.3 Leistung (Performance)29
4.4 Unterstützbarkeit (Supportability)29
4.5 Sonstige Einschränkungen29
5 Risikoakzeptanz30
6 Skizze der Gesamtsystemarchitektur30
7 Lieferumfang30
8 Abnahmekriterien30
9 Glossar31
10 Abkürzungsverzeichnis32
11 Literaturverzeichnis32
12 Abbildungsverzeichnis32

1 Einleitung

Dieses Dokument enthält alle an das zu entwickelnde System gestellten Anforderungen. Die Gliederung orientiert sich am Aufbau des V-Modell-XT®[1]-Produkts „Anforderungen (Lastenheft)“, ist jedoch zur Verwendung für die Veranstaltung „Software-Projekte“ in Informatik-Curricula der OTH-Amberg-Weiden angepasst worden (und nicht konform zum V-Modell-XT): Teilnehmer dieser Veranstaltung erhalten von ihrem „Auftraggeber“ lediglich einen Überblick über das gewünschte System, was ungefähr dem Thema „Ausgangssituation und Zielsetzung“ in diesem Dokument entspricht; die Anforderungen müssen die Teilnehmer dann in enger Abstimmung mit ihrem „Auftraggeber“ selbst erarbeiten und in diesem Dokument niederlegen. Dadurch sollen sie Gelegenheit erhalten, auch Tätigkeiten der System-Analyse intensiver zu üben. Die „Auftraggeberseite“ liefert also nicht – wie im V-Modell-XT vorgesehen - das komplette Lastenheft, aus dem die „Auftragnehmer Seite“ ein separates Pflichtenheft ableitet; stattdessen wird das hier vorliegende Dokument vom studentischen Entwicklerteam zur Dokumentation der Analyse-Ergebnisse erstellt und zugleich als Ersatz für die im V-Modell-XT vorgesehenen Dokumente Lasten- und Pflichtenheft verwendet.

Kern dieses Dokuments sind die funktionalen und nicht-funktionalen Anforderungen an das System, sowie eine Skizze des Gesamtsystementwurfs. Der Entwurf berücksichtigt die zukünftige Umgebung und Infrastruktur, in der das System später betrieben wird, und gibt Richtlinien für Technologieentscheidungen. Ebenfalls Teil der Anforderungen ist die Festlegung von Lieferbedingungen und Abnahmekriterien.

Die funktionalen und nicht-funktionalen Anforderungen dienen nicht nur als Vorgaben für die Entwicklung, sondern sind zusätzlich Grundlage der Anforderungsverfolgung und des Änderungsmanagements. Die Anforderungen sollten so aufbereitet sein, dass die Verfolgbarkeit (Traceability) sowie ein geeignetes Änderungsmanagement für den gesamten Lebenszyklus eines Systems möglich sind.

Im Allgemeinen sollten keine technischen Lösungen vorgegeben werden, um Architekten und Entwickler bei der Suche nach optimalen technischen Lösungen nicht einzuschränken.

Zur besseren Lesbarkeit wird im vorliegenden Dokument auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Es wird das generische Maskulinum verwendet, wobei beide Geschlechter gleichermaßen gemeint sind.

2 Ausgangssituation und Zielsetzung

Hauptthema ist die Anwendung des BB-84 Protokolls zum besseren Erlernen dieses Protokolls. Es gibt noch keine Anwendung, die das spielerische Erlernen vollumfänglich ermöglicht. Hier setzen wir an und entwickeln eine Lernplattform in der Form eines kleinen Spiels, welches dem Nutzer ermöglicht, alle Facetten des BB-84 Protokolls zu verstehen. Dies beinhaltet auch, dass ein oder mehrere Anwender Fehler machen können und am Ende eine Bewertung erhalten, aus welcher sie lernen können. Mit diesen Features soll es auch für jüngere Nutzer geeignet sein und für Unterrichtszwecke verwendbar sein. Ein beispielhafter korrekter Ablauf sowie zwei mögliche Fehlerszenarien liegen als Sequenzdiagramme in dem Dokument „Sequenzdiagramme.pdf“ bei.
Primärer Kunde ist die OTH Amberg Weiden, im speziellen Prof. Dr. Hoffman der Fakultät EMI. Es ist geplant die Software für Demonstrationszwecke einzusetzen. Der Einsatz dieser Software ist für das Wintersemester 2021/22 geplant. Da die Software noch auf die Rechner der OTH eingepflegt werden muss, wird noch etwas Vorlauf benötigt, woraus sich die Deadline Ende des Sommers 2021 ergibt. Durch das spezielle IT-Ökosystem der Hochschule, in welchem das Programm nicht nur lauffähig, sondern auch kompilierbar sein soll, ergeben sich einige technische Einschränkungen. Das Zielsystem ist ein Windows 10 Enterprise LTSC 2019 Rechner. Auf diesem Rechner ist Visual Studio Enterprise 2019 in der Version 16.6.1, Eclipse IDE for Java Developers, Version: 2020-06(4.16.0) und der Browser Firefox vorinstalliert. Ebenso ist das .NET Framework in der Version 4.7.03190 verfügbar. Daraus ergeben sich folgende Entwicklungsplattformen: C++ oder C# mit dem .NET Framework, oder Java mit der JDK-Version 14.0.1. Das Entwicklerteam hat sich einstimmig auf C# mit dem .NET Framework in der Version 4.7.03190 entschieden. Zur Versionsverwaltung wird ein GitLab-Server der OTH-Verwendet.

1 V-Modell® ist eine geschützte Marke der Bundesrepublik Deutschland.

Der Name des Projekts setzt sich aus zwei Teilen zusammen. Der erste Teil „QuantumCrypto“, ist eine Abkürzung für Quanten Kryptographie. Der zweite Teil „Cram“, ist eine umgangssprachliche Bezeichnung für intensives Lernen. Der Name bedeutet also sich intensiv mit Methoden der Quanten Kryptographie zu beschäftigen. In diesem Fall mit dem BB-84-Protokoll. Im Folgenden wird für „QuantenCryptoCram“ mit QCC abgekürzt.

3 Funktionale Anforderungen

3.1 Use-Case Übersicht

- UC1: Nutzer beendet Programm
- UC2: Lokaler Modus Starten
- UC3: Simulation starten (Lokal)
- UC4: Zurück zum Hauptmenü
- UC5: Rolle auswählen
- UC6: Lernhilfe anzeigen
- UC7: “Nachricht verschlüsseln (Alice)” - Oberfläche anzeigen
- UC8: “Nachricht entschlüsseln (Eve)” - Oberfläche anzeigen
- UC9: “Nachricht empfangen (Bob)” - Oberfläche anzeigen
- UC10: Rückkehr zur Simulationsübersicht
- UC11: Photonen Konfiguration erzeugen
- UC12: Photonen übertragen
- UC13: Polarisation zum Empfangen wählen
- UC14: Polarisation senden
- UC15: Polarisation vergleichen
- UC16: Übereinstimmungen der Polarisation mitteilen
- UC17: Schlüsselbit im PreKey zum Vergleich auswählen
- UC18: Ausgewählte PreKey-Bits senden
- UC19: Empfangene PreKey-Bits mit Eigenen vergleichen
- UC20: Antwort zum PreKey-Bit Vergleich
- UC21: Finale Auswahl fertig (Alice)
- UC22: Messen fertig (Eve)
- UC23: Finale Auswahl fertig (Bob)
- UC24: Nachricht verfassen
- UC25: Nachricht senden
- UC26: FinalKey-Bits bearbeiten (Eve)
- UC27: Simulation beenden und Nutzerbewertung ausgeben
- UC28: Passwort für Rolle einstellen

Ein detailliertes Use-Case Diagramm befindet sich im beiliegenden Dokument „UC-Diagram.pdf“.

3.2 Use-Case Beschreibungen

Nutzer beendet Programm	
Kennung	UC-1
Priorität	Hoch
Kurzbeschreibung:	
Der Nutzer beendet das Programm.	
Vorbedingung(en):	
Der Nutzer befindet sich in der “Hauptmenü”-Oberfläche.	
Nachbedingung(en):	
Das Programm hat sich korrekt beendet.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den “Anwendung beenden”-Button. 2. Das Programm beendet sich. **Ende**.
Ablauf-Varianten:	
1a	Der Nutzer drückt die Tastenkombination Alt+F4.
	1. Das System fängt das Terminierungssignal ab. **Rückkehr nach**: 2
Spezielle Anforderungen:	
Zu klärende Punkte:	

Lokaler Modus auswählen	
Kennung	UC-2
Priorität	Hoch
Kurzbeschreibung:	
Der Nutzer wählt die Option „Lokal“ aus, um nur an einem Computer zu simulieren	
Vorbedingung(en):	
Es darf keine Simulation gestartet sein. Der Nutzer befindet sich in der “Hauptmenü”-Oberfläche.	
Nachbedingung(en):	
Der Nutzer befindet sich in der “Lokaler Modus Spieloptionen”-Oberfläche.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den “Lokaler Modus Starten“- Button. 2. Das System wechselt in die “Lokaler Modus Spieloptionen“-Oberfläche. **Ende**.
Ablauf-Varianten:	
1a	Nutzer betätigt den „Zurück-Pfeil“-Button
	1. Der Nutzer betätigt den „Zurück-Pfeil“-Button. 2. Das System kehrt in die „Hauptmenü“-Oberfläche zurück. **Ende.**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Simulation starten (Lokal)	
Kennung	UC-3
Priorität	Hoch
Kurzbeschreibung:	
Die Simulation des BB884-Protokolls wird gestartet. Die Rollenauswahl wird aktiviert.	
Vorbedingung(en):	
Es darf noch keine Simulation gestartet sein. Der Nutzer befindet sich in der „Lokaler Modus Spieloptionen“-Oberfläche.	
Nachbedingung(en):	
Sobald die Simulation gestartet ist, sind keine Änderungen in den Optionen mehr möglich. Der Nutzer befindet sich in der „Simulationsübersicht“-Oberfläche.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den „Simulation starten“- Button. 2. Die Checkbox „Eve aktivieren“ ist nicht betätigt und das System wechselt in die „Simulationsübersicht“-Oberfläche ohne, dass Eve angezeigt wird. **Ende**.
Ablauf-Varianten:	
2a	Die Checkbox „Eve aktivieren“ ist betätigt.
	1. Das System wechselt in die „Simulationsübersicht“-Oberfläche mit Eve aktiv. 2. Eve wird in der „Simulationsübersicht“-Oberfläche aktiviert und somit eingeblendet. **Ende**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Zurück zum Hauptmenü	
Kennung	UC-4
Priorität	Hoch
Kurzbeschreibung:	
Eine gestartete Simulation soll in der Simulationsübersicht jederzeit abgebrochen werden können, sodass sich der Nutzer wieder in der "Hauptmenü"-Oberfläche befindet. Auch aus den verschiedenen Einstellungsmenüs soll der Nutzer in das Hauptmenü zurückkehren können.	
Vorbedingung(en):	
Der Nutzer befindet sich einem der drei Oberflächen: „Simulationsübersicht", „Lokaler Modus Spieloptionen" oder „Netzwerkmodus".	
Nachbedingung(en):	
Die Simulation ist beendet.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den „Zurück zum Hauptmenü"- Button. 2. Das System fordert den Nutzer auf zu bestätigen. 3. Die Simulation wird abgebrochen. 4. Das System kehrt zur "Hauptmenü"-Oberfläche zurück. **Ende**.
Ablauf-Varianten:	
2a	Der Nutzer verneint Bestätigung.
	1. Das System kehrt in die „Simulationsübersicht"-Oberfläche zurück **Ende.**
2b	Der Nutzer befindet sich in der Oberfläche "Lokaler Modus Spieloptionen"
	1. Das System übernimmt alle getroffenen Einstellungen. **Rückkehr nach: 4**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Rolle auswählen	
Kennung	UC-5
Priorität	Hoch
Kurzbeschreibung:	
Ein Nutzer wählt in der „Simulationsübersicht“-Oberfläche seine Rolle im Protokoll aus, um mit seinem Teil des Protokolls fortzufahren.	
Vorbedingung(en):	
Die Simulation wurde gestartet. Der Button „Schlüssel fertig“ (Alice. Bob) bzw. “Messen beenden” (Eve) wurde noch nicht betätigt. Der Nutzer befindet sich der “Simulationsübersicht”-Oberfläche.	
Nachbedingung(en):	
Der Nutzer befindet sich in der Oberfläche der gewählten Rolle.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den entsprechenden Button einer der verfügbaren Rollen (Alice, Bob, ggf. Eve). 2. Das System zeigt die Oberfläche der gewählten Rolle an. **Ende**.
Ablauf-Varianten:	
2a	Nutzer wählt die Rolle “Alice”.
	1. Das System überprüft, ob ein Passwort für die Rolle „Alice“ eingestellt ist. 2. (optional) Das System fragt das Passwort für die Rolle „Alice“ ab. 3. Das System wechselt zur „Alice“-Oberfläche. **Ende.**
2b	Nutzer wählt die Rolle “Bob”.
	1. Das System überprüft, ob ein Passwort für die Rolle „Bob“ eingestellt ist. 2. (optional) Das System fragt das Passwort für die Rolle „Bob“ ab. 3. Das System wechselt zur „Bob“-Oberfläche. **Ende.**
2c	Nutzer wählt die Rolle “Eve”.
	1. Das System überprüft, ob ein Passwort für die Rolle „Eve“ eingestellt ist. 2. (optional) Das System fragt das Passwort für die Rolle „Eve“ ab. 3. Das System wechselt zur „Eve“-Oberfläche. **Ende.**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Lernhilfe anzeigen	
Kennung	UC-6
Priorität	Niedrig
Kurzbeschreibung:	
Der Nutzer erhält Hilfestellungen für die aktuelle Oberfläche und relevante Informationen zum BB84-Protokoll angezeigt.	
Vorbedingung(en):	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer betätigt ein „Lernhilfen“-Icon. 2. Die Simulation pausiert. 3. Das System öffnet ein Lernhilfe-Popup-Fenster für die aktuelle Oberfläche. 4. Der Nutzer schließt das Fenster über das Standard „Schließen“-Icon oben rechts. 5. Die Simulation wird fortgesetzt. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

“Nachricht verschlüsseln (Alice)” - Oberfläche anzeigen	
Kennung	UC-7
Priorität	Mittel
Kurzbeschreibung:	
Der Nutzer wechselt in die „Nachricht verschlüsseln (Alice)“-Oberfläche.	
Vorbedingung(en):	
Der Nutzer befindet sich in der „Simulationsübersicht“-Oberfläche. Der Button „Nachricht versenden“ ist aktiviert.	
Nachbedingung(en):	
Das System befindet sich in der „Nachricht verschlüsseln (Alice) “-Oberfläche.	
Normaler Ablauf:	
	1. Der Nutzer klickt den Button „Nachricht versenden“. 2. Das System überprüft ob für die Rolle Alice ein Passwort eingestellt ist. 3. (optional) Das System fragt das Passwort für die Rolle Alice ab 4. Das System wechselt in das „Nachricht verschlüsseln Alice“-Oberfläche. 5. Das System zeigt dem Nutzer erneut den aktuellen FinalKey übersichtlich an. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

“Nachricht entschlüsseln (Eve)” - Oberfläche anzeigen	
Kennung	UC-8
Priorität	Mittel
Kurzbeschreibung:	
Der Nutzer wechselt in die „Nachricht entschlüsseln (Eve)“-Oberfläche.	
Vorbedingung(en):	
Der Nutzer befindet sich in der „Simulationsübersicht“-Oberfläche. Der Button „Nachricht entschlüsseln“ ist aktiviert. Eve hat eine Ciphernachricht über den öffentlichen Kanal erhalten und im Notebook eingetragen.	
Nachbedingung(en):	
Das System befindet sich in der „Nachricht entschlüsseln (Eve)“-Oberfläche.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den Button „Nachricht entschlüsseln “. 2. Das System überprüft ob für die Rolle Eve ein Passwort eingestellt ist. 3. (optional) Das System fragt das Passwort für die Rolle Eve ab 4. Das System wechselt in die „Nachricht entschlüsseln (Eve)“ Oberfläche. 5. Das System erzeugt eine bearbeitbare Liste, die Eves unbearbeiteten FinalKey enthält. 6. Das System zeigt dem Nutzer alle nützlichen Informationen an, die Eve durch das Messen erlangen konnte. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

“Nachricht empfangen (Bob)” - Oberfläche anzeigen	
Kennung	UC-9
Priorität	Mittel
Kurzbeschreibung:	
Der Nutzer wechselt in die „Nachricht empfangen (Bob)“-Oberfläche.	
Vorbedingung(en):	
Bob hat eine Ciphernachricht über den öffentlichen Kanal erhalten und im Notebook eingetragen. Der Nutzer befindet sich in der „Simulationsübersicht“-Oberfläche. Der Button „Nachricht empfangen“ ist aktiviert.	
Nachbedingung(en):	
Das System wechselt in die „Nachricht empfangen (Bob)“-Oberfläche	
Normaler Ablauf:	
	1. Der Nutzer klickt den „Nachricht empfangen“-Button. 2. Das System überprüft ob für die Rolle Bob ein Passwort eingestellt ist. 3. (optional) Das System fragt das Passwort für die Rolle Bob ab 4. Das System wechselt in das „Nachricht empfangen (Bob)“-Oberfläche. 5. Das System zeigt den von Bob ermittelten „FinalKey“ und den Ciphertext an. 6. Das System entschlüsselt die Ciphernachricht mit den von Bob ermittelten „FinalKey“. 7. Das System zeigt die entschlüsselte Nachricht als ASCII-Text in einem Textfeld an. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

Rückkehr zur Simulationsübersicht	
Kennung	UC-10
Priorität	Hoch
Kurzbeschreibung:	
Die Ansicht kehrt zurück zur Simulationsübersicht.	
Vorbedingung(en):	
Der Nutzer befindet sich in der Oberfläche einer Rolle.	
Nachbedingung(en):	
Der Nutzer befindet sich in der „Simulationsübersicht“-Oberfläche. Alle bisher erstellten Notebook-Einträge sind gespeichert.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den „Zurück”-Button 2. Das System speichert die aktuellen Notebook-Einträge der entsprechenden Rolle. 3. Das System wechselt zur „Simulationsübersicht“-Oberfläche. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

Photonen Konfiguration erzeugen	
Kennung	UC-11
Priorität	Hoch
Kurzbeschreibung:	
Alice erzeugt Datenbits und das dazugehörige Polarisationsschema.	
Vorbedingung(en):	
Der Nutzer befindet sich in der “Alice”-Oberfläche.	
Nachbedingung(en):	
Die Daten stehen in Alice’s Notebook.	
Normaler Ablauf:	
	1. (Optional) Alice wählt ein Datenbit aus. 2. (Optional) Alice wählt eine Polarisation. 3. Alice betätigt mit dem „Ein Photon übernehmen“-Button. 4. Das System fügt die Auswahl in die “Datenbit”- und “Polarisation”-Spalten von Alice’s Notebook mit einer eindeutigen ID hinzu. **Ende**.
Ablauf-Varianten:	
1a	Automatisch n zufällige Konfigurationen erzeugen
	1. Der Nutzer gibt die Anzahl der zufällig zu erstellenden Photonen an. 2. Der Nutzer betätigt den „Zufällige Photonen übernehmen“-Button. 3. Das System erzeugt die eingegebene Anzahl an zufällig polarisierten Photonen. **Rückkehr nach**: 4
Spezielle Anforderungen:	
Zu klärende Punkte:	

Photonen übertragen	
Kennung	UC-12
Priorität	Hoch
Kurzbeschreibung:	
Alice überträgt für alle noch nicht gesendeten Bit-Polarisationskombinationen je ein entsprechendes Photon über den Quantenkanal.	
Vorbedingung(en):	
Der Nutzer befindet sich im Alice-Menü. Es existieren noch nicht gesendeten Photonen im Alice-Notebook.	
Nachbedingung(en):	
Alle Photonen im Alicenotebook sind als “Gesendet” markiert	
Normaler Ablauf:	
	1. Alice betätigt den „Photonen Senden“-Button. 2. Das System überträgt die noch nicht gesendeten Photonen über den Quantenkanal und markiert diese als “Gesendet”. **Ende**.
Ablauf-Varianten:	
2a	Es existieren festgelegte Polarisationen zum Messen bei den Empfängern (Eve, Bob)
	1. Das System misst (/klont) automatisch die ankommenden Photonen mit dem im Empfänger-Notebook festgelegten Polarisationsschema. 2. Das Messergebnis wird vom System in die Spalte “Datenbit” im Notebook des Empfängers eingetragen. **Ende.**
2b	Es sind schon Übereinstimmungen der Polarisationen bei Bob markiert (Fehler im Protokollablauf)
	1. Das System misst (/klont) automatisch die ankommenden Photonen mit dem im Empfänger-Notebook festgelegten Polarisationsschema. 2. Das Messergebnis wird vom System in die Spalte “Datenbit” im Notebook des Empfängers eingetragen. 3. Das System trägt die markierten Bits in die PreKey und FinalKey Spalte von Bob ein. **Ende**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Polarisation zum Empfangen wählen	
Kennung	UC-13
Priorität	Hoch
Kurzbeschreibung:	
Der Nutzer wählt als Bob oder Eve die Polarisation zum Empfangen eines Photons.	
Vorbedingung(en):	
Das System befindet sich in der Rollenoberfläche von Bob oder Eve.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer wählt die Polarisation aus. 2. Die gewählte Polarisation wird in die „Polarisation“-Spalte des Notebooks der aktuellen Rolle eingetragenen. 3. (Optional) Messen **Ende**.
Ablauf-Varianten:	
1a	N Polarisationen zufällig wählen
	1. Anzahl der zu messenden Photonen eingeben. 2. Button „Erzeugen“ betätigen. **Rückkehr nach: 2**
1b	Eve hat geleakte Polarisationen von Bob oder Alice vor den Photonen empfangen.
	1. Eve kann diese Polarisationen mit jeweils mit einem „Übernehmen“-Button zum Messen der Photonen auswählen. 2. Das System übernimmt die ausgewählten Polarisationen für noch nicht gemessene Photonen. **Rückkehr nach: 3**
3a	Es stehen Photonen auf den Quantenkanal an.
	1. Das System misst (/klont) die verfügbaren Photonen automatisch mit der festgelegten Polarisation. 2. Das System trägt das Ergebnis in die Spalte des Notebooks ein. **Ende**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Polarisation senden	
Kennung	UC-14
Priorität	Hoch
Kurzbeschreibung:	
Alice oder Bob senden ihre Polarisationen an den jeweils anderen über den öffentlichen Datenkanal.	
Vorbedingung(en):	
Das System befindet sich in der Rollenoberfläche von Alice oder Bob. Es sind bereits Photonen gesendet oder empfangen worden.	
Nachbedingung(en):	
Jeder Teilnehmer kennt die gesendeten Polarisationen.	
Normaler Ablauf:	
	1. Alice oder Bob drücken den „Polarisation Senden“-Button. 2. Alle im Notebook eingetragenen, noch nicht gesendeten Polarisationen werden über den öffentlichen Kanal gesendet. 3. Die Polarisationen werden als gesendet markiert. 4. Die Polarisationen werden jeweils in die Listen der Empfänger (Alice/Bob und Eve) eingetragen. **Ende**.
Ablauf Variante:	
4a	Es wird mit Eve gespielt
	1. Eve markiert ihre Übereinstimmungen farbig. **Ende**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Polarisation vergleichen	
Kennung	UC-15
Priorität	Hoch
Kurzbeschreibung:	
Alice oder Bob vergleichen die Polarisation, welche sie von der jeweils anderen Partei erhalten haben.	
Vorbedingung(en):	
Das System befindet sich in der Rollenoberfläche von Alice oder Bob. Polarisation wurden veröffentlicht.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer vergleicht, die empfangen Polarisation mit der Eigenen. 2. Bei einer Übereinstimmung klickt der Nutzer die entsprechende Checkbox an. 3. Das System hebt die markierten Zeilen farbig. 4. Das System überträgt das zugehörige Datenbit in die PreKey und FinalKey Spalte des Notebooks der Rolle. **Ende**.
Ablauf-Varianten:	
2a	Der Nutzer klickt auf den Button “Auto-Check”.
	1. Das System vergleicht die Polarisationen automatisch und bei Übereinstimmung aktiviert es die Checkbox. **Rückkehr nach: 3**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Übereinstimmungen der Polarisation mitteilen	
Kennung	UC-16
Priorität	Hoch
Kurzbeschreibung:	
Alice oder Bob teilen jeweils die Übereinstimmungen in der Polarisation über den öffentlichen Datakanal mit. Aus den Übereinstimmungen ergibt sich der gemeinsame PreKey.	
Vorbedingung(en):	
Das System befindet sich in der Rollenoberfläche von Alice oder Bob. Polarisationsbits der anderen Rolle sind markiert worden (UC-19).	
Nachbedingung(en):	
PreKey-Liste bei Alice und Bob ist erstellt. Die Übereinstimmungen wurden auf dem öffentlichen Datenkanal übertragen.	
Normaler Ablauf:	
	1. Alice oder Bob klickt auf den Button "Übereinstimmungen senden". 2. Alice oder Bob sendet die Positionen, an denen eine Übereinstimmung markiert wurde über den öffentlichen Kanal an alle anderen Teilnehmer. 3. Das System hebt beim Empfänger die Übereinstimmungen visuell hervor. 4. Das System trägt die markierten Bits in die PreKey und FinalKey Spalte aller Teilnehmer ein. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

Schlüsselbit im PreKey zum Vergleich auswählen	
Kennung	UC-17
Priorität	Hoch
Kurzbeschreibung:	
Alice oder Bob wählt Bits des PreKey zum Vergleich aus und sendet diese.	
Vorbedingung(en):	
Es liegt ein PreKey bei der gewählten Rolle vor (UC-19/UC-20). Das System befindet sich in der Rollenoberfläche von Alice oder Bob.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer wählt ein Bit im PreKey aus. 2. Das System markiert das Bit. **Ende**.
Ablauf-Varianten:	
1a	Anzahl zufälliger Bits zum PreKey-Vergleich auswählen
	1. Der Nutzer gibt eine Zahl in das „Anzahl“-Feld ein. 2. Der Nutzer betätigt den „Zufällige PreKey-Bits übernehmen“-Button. 3. Das System demarkiert alle noch nicht gesendeten schon markierten PreKey-Bits 4. Das System wählt zufällig die eingegebene Anzahl in der Spalte „Eigene PreKey Auswahl“ an. **Ende.**
2a	Die Position ist schon ausgewählt
	1. Das System demarkiert das Bit. **Ende.**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Ausgewählte PreKey-Bits senden	
Kennung	UC-18
Priorität	Hoch
Kurzbeschreibung:	
Es werden die Positionen und Werte der ausgewählten PreKey-Bits übertragen.	
Vorbedingung(en):	
Das System befindet sich in der Rollenoberfläche von Alice oder Bob. Es muss mindesten ein PreKey-Bit ausgewählt sein.	
Nachbedingung(en):	
Werte und Position der PreKey-Bits liegen auf dem öffentlichen Kanal.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den Button „PreKey Auswahl senden“. 2. Das System kennzeichnet alle ausgewählten PreKey-Bits im eigenen Notebook als “Gesendet” und sperrt die entsprechenden “Auswahl“-Buttons. 3. Das System überträgt alle ausgewählten PreKey-Bits mit Positionen auf den öffentlichen Datenkanal und entfernt die entsprechenden Bits aus der “FinalKey”-Spalte des eigenen Notebooks (da Bits zum PreKey-Vergleich nicht in den FinalKey wandern/ bzw. verworfen werden). 4. Das System trägt die gesendeten PreKey-Bits in die Spalte “PreKey Auswahl Alice/Bob” der Notebooks der anderen Teilnehmer ein und entfernt ebenfalls deren entsprechenden Bits aus der “FinalKey”-Spalte. 5. Das System veranschaulicht dem Nutzer in der Eve Rolle, dass Alice und Bob das entsprechende PreKey-Bit (bedingt durch den Vergleich) verworfen haben. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

Empfangene PreKey-Bits mit Eigenen vergleichen	
Kennung	UC-19
Priorität	Hoch
Kurzbeschreibung:	
Das System oder der Nutzer vergleicht die empfangenen PreKey-Bits mit den Bits seines PreKeys.	
Vorbedingung(en):	
Die Rolle ist Alice oder Bob. Das eigene Notebook enthält Einträge in der eigenen PreKey-Spalte und der PreKey Auswahl Spalte des Anderen.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer klickt bei PreKey-Bits die entsprechende "Übereinstimmung"-Checkbox. 2. Das System hebt die angewählten PreKey-Bits visuell hervor. **Ende**.
Ablauf-Varianten:	
1a	Der Nutzer wählt „Auto Check"
	1. Der Nutzer klickt den „Auto Check"-Button. 2. Das System vergleicht die Bits in den zwei PreKey-Spalten des eigenen Notebooks. 3. Das System wählt die Übereinstimmungen aus. **Rückkehr nach: 2**
2a	Checkbox bereits angewählt
	1. Wenn das PreKey-Bit bereits ausgewählt war, dann wird die visuelle Markierung zurückgenommen. **Ende**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Antwort zum PreKey-Bit Vergleich	
Kennung	UC-20
Priorität	Hoch
Kurzbeschreibung:	
Alice/Bob senden jeweils das Ergebnis des eigenen PreKey-Bitvergleiches an die andere Protokollpartei.	
Vorbedingung(en):	
Der Nutzer befindet sich in der Alice bzw. Bob-Oberfläche. Es wurden PreKey-Bits empfangen.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer betätigt den Button „PreKey Übereinstimmungen senden". 2. Das System sendet die Positionen der Übereinstimmungen über den öffentlichen Kanal. 3. Das System trägt die Übereinstimmungen im Notebook der anderen Teilnehmer ein. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

Finale Auswahl fertig (Alice)	
Kennung	UC-21
Priorität	Hoch
Kurzbeschreibung:	
Alice teilt dem System mit, dass sie alle Aktionen zur Bestimmung des FinalKey abgeschlossen hat.	
Vorbedingung(en):	
Der Nutzer befindet sich in der Alice-Oberfläche. Es existiert mindestens ein Eintrag in der FinalKey-Spalte in Alice Notebook.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer bestätigt die den FinalKey mit dem „Finale Auswahl fertig“ Button. 2. Das System blockiert alle Nutzeraktionen in der Oberfläche. 3. Das System deaktiviert alle Buttons von Bob, welche Daten an Alice senden. 4. Das System aktiviert den „Messen fertig“-Button bei Eve. 5. Das System aktiviert den "Nachricht versenden"-Button in der "Simulationsübersicht"-Oberfläche **Ende**.
Ablauf Varianten:	
4a	Es wird ohne Eve simuliert
	1. Das System aktiviert den „Finale Auswahl fertig“ Button bei Bob. **Rückkehr nach: 4**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Messen fertig (Eve)	
Kennung	UC-22
Priorität	Hoch
Kurzbeschreibung:	
Eve teilt dem System mit, dass alle Photonen gemessen sind.	
Vorbedingung(en):	
Das System befindet sich in der Rollenoberfläche von Eve. Alice hat den „Auswahl fertig“-Button betätigt. Eve hat alle Photonen gemessen.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer bestätigt den Final Key mit dem „Messen beenden“-Button. 2. Das System blockiert alle Nutzeraktionen in der Oberfläche. 3. Das System aktiviert „Finale Auswahl fertig“-Button bei Bob 4. Das System aktiviert den „Nachricht knacken“-Button in der „Simulationsübersicht“-Oberfläche. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

Finale Auswahl fertig (Bob)	
Kennung	UC-23
Priorität	Hoch
Kurzbeschreibung:	
Bob teilt dem System mit, dass er einen kompletten FinalKey bestimmt hat.	
Vorbedingung(en):	
Das System befindet sich in der Rollenoberfläche von Bob. Eve hat den „Messen fertig“-Button betätigt. Es existiert mindestens ein Eintrag in der FinalKey-Spalte in Bobs Notebook.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer bestätigt den Final Key mit dem „Finale Auswahl fertig“ Button. 2. Das System blockiert alle Nutzeraktionen in der Oberfläche. 3. Das System aktiviert den „Nachricht empfangen“-Button in der „Simulationsübersicht“-Oberfläche. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

Nachricht verfassen	
Kennung	UC-24
Priorität	Hoch
Kurzbeschreibung:	
Alice verfasst eine Nachricht und verschlüsselt diese.	
Vorbedingung(en):	
Der Nutzer befindet sich in der „Nachricht verschlüsseln (Alice)“-Oberfläche. Ein FinalKey steht zur Verfügung.	
Nachbedingung(en):	
Eine verschlüsselte Nachricht steht zum Senden bereit.	
Normaler Ablauf:	
	1. Alice klickt auf „Zu verschlüsselnde Nachricht“ und gibt ihre Nachricht im ASCII-Format ein. 2. Das System verschlüsselt die Nachricht mit dem FinalKey und zeigt den Ciphernachricht an. **Ende**.
Ablauf-Varianten:	
2a	Die Nachricht ist länger als der Schlüssel
	1. Das System verwendet den FinalKey mehrfach für die Verschlüsselung. **Ende**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Nachricht senden	
Kennung	UC-25
Priorität	Hoch
Kurzbeschreibung:	
Die Ciphernachricht von Alice wird auf dem öffentlichen Kanal gesendet.	
Vorbedingung(en):	
Es wurde eine Nachricht verfasst und verschlüsselt. Der Nutzer ist in der „Nachricht verschlüsseln (Alice)“ Oberfläche.	
Nachbedingung(en):	
Die verschlüsselte Nachricht ist auf dem öffentlichen Kanal.	
Normaler Ablauf:	
	1. Der Nutzer drückt den Butten „Senden“. 2. Das System versendet die Ciphernachricht auf dem öffentlichen Kanal und trägt diesen im Notebook der anderen Teilnehmer ein. 3. Das System deaktiviert das Eingabefeld sowie den „Senden“-Button. **Ende**.
Spezielle Anforderungen:	
Zu klärende Punkte:	

FinalKey-Bits bearbeiten (Eve)	
Kennung	UC-26
Priorität	Mittel
Kurzbeschreibung:	
Eve kann ihren FinalKey bearbeiten, um die Ciphernachricht zu entschlüsseln.	
Vorbedingung(en):	
Der Nutzer befindet sich in der „Nachricht entschlüsseln Eve“-Oberfläche. Eve hat eine Ciphernachricht über den öffentlichen Kanal erhalten.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Benutzer kann ein einzelnes Bit auf 0, 1 oder „leer“ setzen. 2. Das System überträgt die Änderung in die Spalte „Final Key“. 3. Das System entschlüsselt den Ciphernachricht mit dem bearbeiteten „Final Key“. 4. Das System zeigt die entschlüsselte Nachricht als ASCII-Text in einem Textfeld an. **Ende**.
Ablauf Varianten:	
1a	Zufällige Bits erzeugen.
	1. Der Nutzer betätigt den „Unbekannte Key Bits mit zufälligen Daten füllen“-Button. 2. Alle Unbekannten Bits erhalten zufällige Werte zugewiesen. **Rückkehr nach: 3**
1b	Der Nutzer setzt alle Unbekannten Bits zurück.
	1. Der Nutzer betätigt den „Zurücksetzen“-Button. 2. Alle unbekannten Bits werden entfernt. **Rückkehr nach: 3**
Spezielle Anforderungen:	
Zu klärende Punkte:	

Auswertung anzeigen und Simulation beenden	
Kennung	UC-27
Priorität	Niedrig
Kurzbeschreibung:	
Der Nutzer erhält vom System eine Bewertung des Protokollablaufes. Der Nutzer befindet sich in der „Simulationsübersicht“-Oberfläche.	
Vorbedingung(en):	
Alice, Bob haben den „Finale Auswahl Fertig“-Button betätigt und Eve hat den „Messen Fertig“ betätigt.	
Nachbedingung(en):	
Die Simulation ist beendet und der Nutzer ist im Hauptmenü.	
Normaler Ablauf:	
	1. Der Nutzer betätigt den „Auswertung“-Button 2. Das System berechnet Kenngrößen und zeigt diese mit der "Protokollanalyse"-Oberfläche an. 3. Der Nutzer betätigt den "Zurück zum Hauptmenü"-Button 4. Das System fordert den Nutzer auf zu bestätigen. 5. Der Nutzer bestätigt. 6. Das System wechselt in die Hauptmenü-Oberfläche **Ende**.
Ablauf Variante:	
3a	Der Nutzer betätigt den „Zurück“ Button
	1. Der Nutzer befindet sich wieder in der "Simulationsübersicht"-Oberfläche. Ende
5a	Der Nutzer bestätigt nicht
	1. Der Nutzer befindet sich weiterhin in der "Protokollanalyse"-Oberfläche. Ende
Spezielle Anforderungen:	
Zu klärende Punkte:	

Passwort für Rolle einstellen	
Kennung	UC-28
Priorität	Hoch
Kurzbeschreibung:	
Der Nutzer legt für seine Rolle ein Passwort fest.	
Vorbedingung(en):	
Der Nutzer ist in der “Lokaler Modus Spieloptionen”-Oberfläche.	
Nachbedingung(en):	
Normaler Ablauf:	
	1. Der Nutzer klickt auf einen der „Passwort …“-Buttons. 2. Das System öffnet einen Passwort Eingabedialog. 3. Der Nutzer gibt ein Passwort ein. 4. Der Nutzer bestätigt mit dem „OK“-Button 5. Das System legt das Passwort für diese Rolle fest und schließt den Eingabedialog. 6. Eine Checkbox zeigt an das für diese Rolle ein Passwort eingestellt ist. **Ende**.
Ablauf-Varianten:	
3a	Der Nutzer gibt kein Passwort ein
	1. Der Nutzer kein Passwort ein. 2. Der Nutzer betätigt den „OK“-Button. 3. Das System stellt kein Passwort für die Rolle ein die Checkbox wird nicht betätigt. 4. Der Eingabedialog schließt sich. **Ende.**
4a	Der Nutzer betätigt den „Abbrechen“-Button
	1. Der Nutzer betätigt den „Abbrechen“-Button. 2. Das System übernimmt nichts aus dem Eingabedialog. 3. Der Eingabedialog schließt sich. **Ende.**
5a	Es ist schon ein Passwort festgelegt
	1. Das vom Nutzer neu eingebe Passwort überschreibt das Alte. 2. Der Eingabedialog schließt sich. **Rückkehr nach: 6**
Spezielle Anforderungen:	
Zu klärende Punkte:	

3.3 (Sonstige) Funktionalität

ID	Beschreibung	Querverweise
FR-001	Das System soll im Fehlerfall eine passende Meldung ausgeben	

3.4 Modell des Problembereichs (Konzeptionelles Datenmodell)

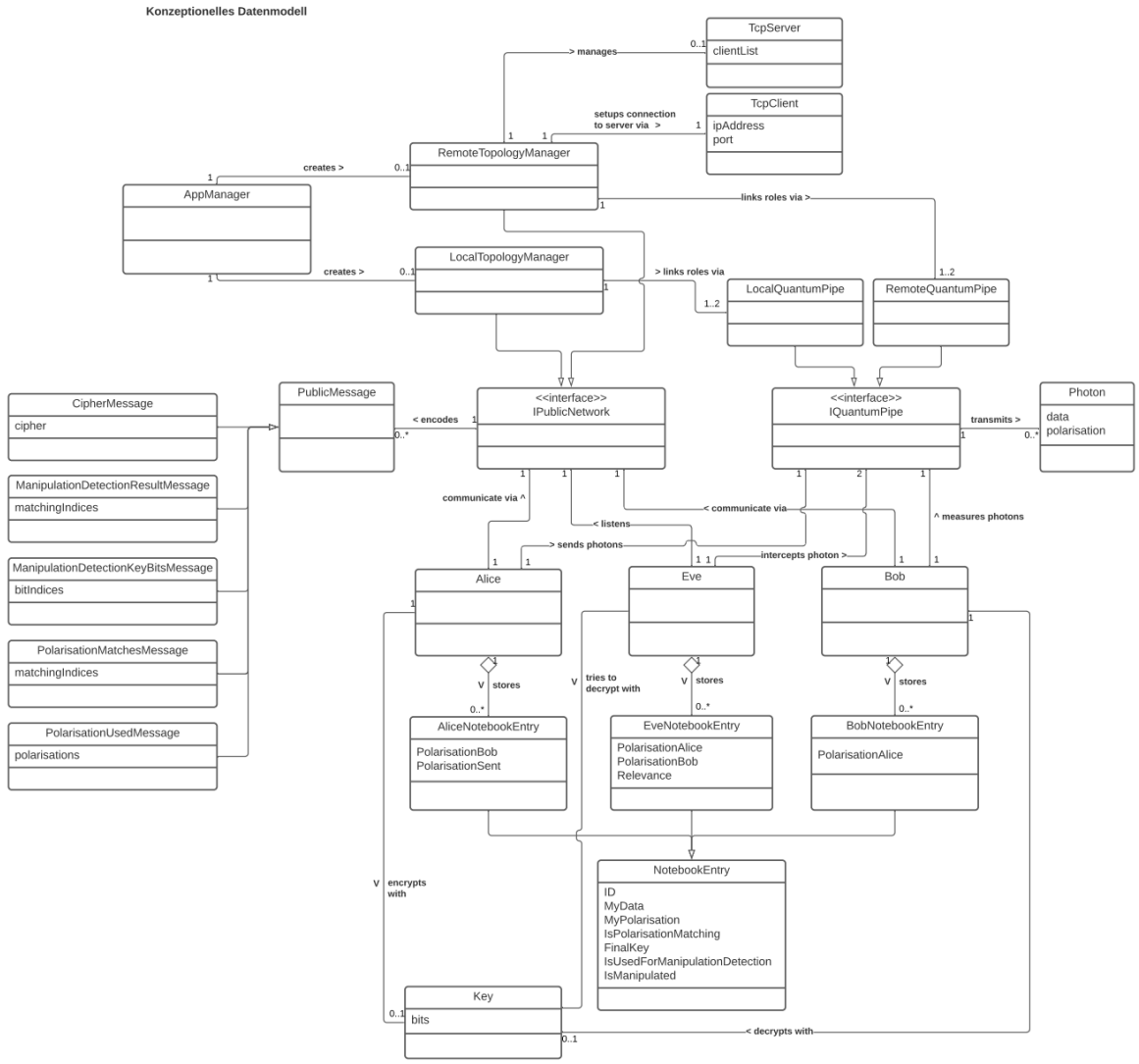

Abbildung 1: Konzeptionelles Datenmodell

Aufgrund der verschiedenen Anforderungen an die unterschiedlichen Rollen Alice, Bob und Eve wurden diese in drei separate Klassen unterteilt. Da jeder für das Protokoll diverse Informationen speichern muss, hat jeder Teilnehmer eine Art Datenbank mit „NotebookEntry“-Elementen, die sich von Rolle zu Rolle unterscheiden können. Um die Kommunikation zwischen den Parteien hinsichtlich Netzwerkfähigkeit und der optionalen Natur von Eve zu ermöglichen, werden die Zwischenbeziehungen abstrahiert. Dazu wird jeder Rolle sowohl ein Objekt, welches das „IPublicNetwork“-Interface implementiert, als auch eines mit „IQuantumPipe“-Interface übergeben. Das „IPublicNetwork“ repräsentiert hierbei die konventionelle, ungesicherte Kommunikation zwischen allen Teilnehmern. Bei der „IQuantumPipe“ hingegen handelt es sich um eine Repräsentation des Quantenkanals auf welchem sich die Quanteneffekte zur Sicherung zunutze gemacht werden und verbindet jeweils nur zwei Personen. Ist Eve vorhanden, so liegt demnach eine Pipe zwischen Alice und Eve und eine weitere zwischen Eve und Bob. Andernfalls

werden Alice und Bob über nur eine Pipe unmittelbar verbunden. Befinden sich alle Teilnehmer gemeinsam an einem Rechner, so findet der Datenaustausch programmintern unter Verwendung eines „Event-Subscriber-Models“ statt. Verteilen sich die Nutzer jedoch auf mehrere, im Netzwerk verknüpfte Rechner, werden im Hintergrund andere Implementierungen der „IPublicNetwork“ und „IQuantumPipe“-Interfaces eingesetzt. Diese geben die zu delegierenden Informationen sofern nötig an einen „TCPClient“ weiter, welcher seinerseits die Daten über das Netzwerk an einen zentralen „TCPServer“ überträgt. Dieser wird von nur einem Teilnehmer erzeugt und übernimmt das Zustellen der Informationen an die korrekte Endstation, bei welcher die Daten dann nahtlos über die entsprechenden Schnittstellen wieder ausgelesen werden können. Diese Infrastruktur ermöglicht eine komplett von der tatsächlichen Verteilung der Nutzer unabhängige Entwicklung des restlichen Systems.

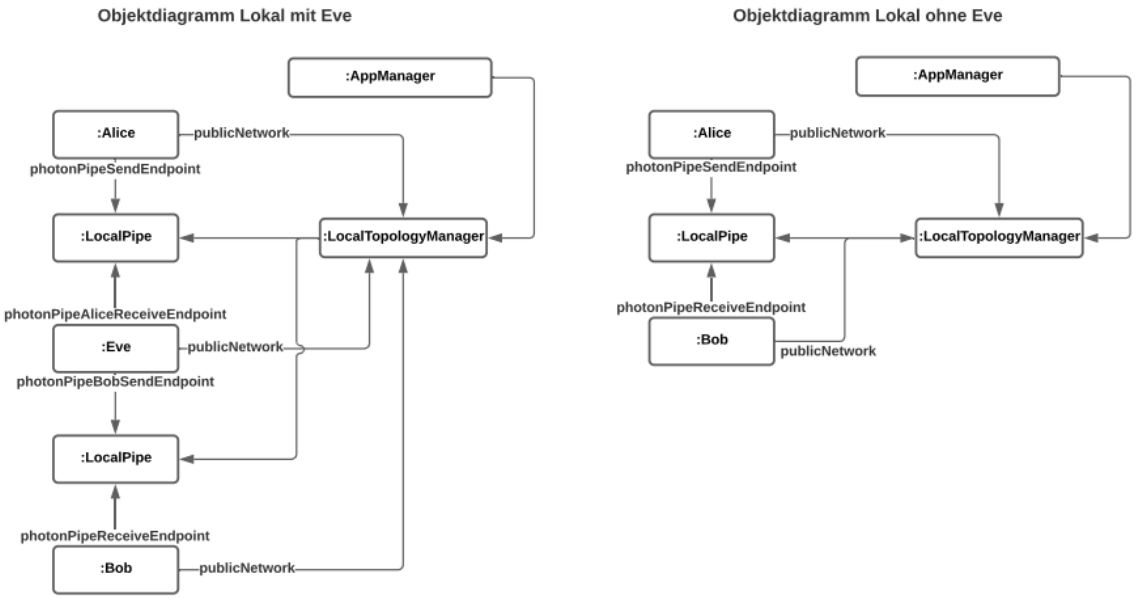

Abbildung 2: Objektdiagramm Lokales Spiel (links: mit Eve; rechts: ohne Eve)

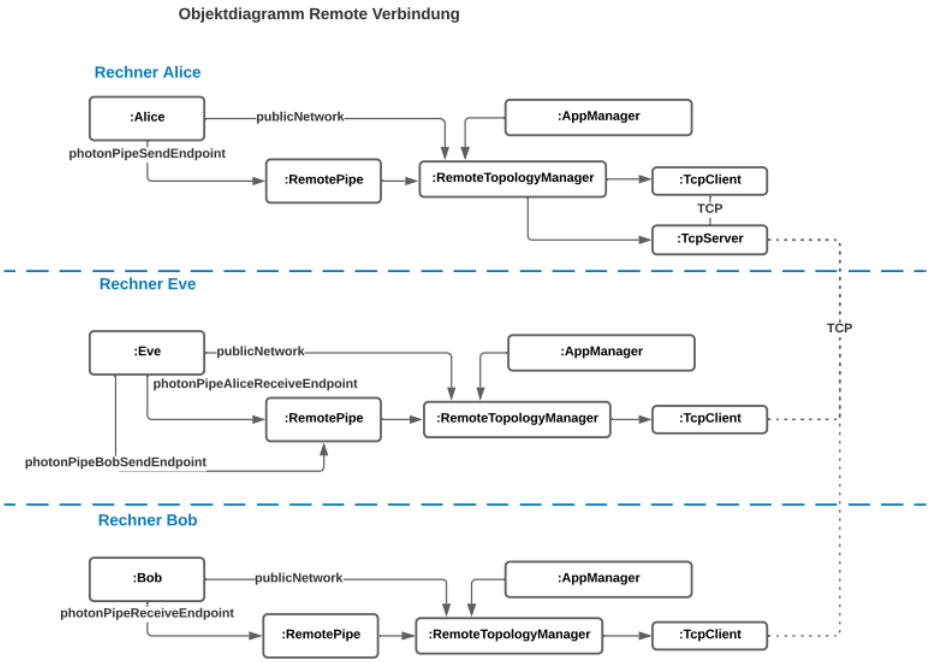

Abbildung 3: Objektdiagramm Remote Spiel

4 Nicht-Funktionale Anforderungen

4.1 Benutzbarkeit (Usability)

ID	Beschreibung	Querverweise
UR-001	Das System soll einfach zu verstehende UI Elemente zur Verfügung stellen.	
UR-002	Das System soll intuitiv bedienbar sein.	
UR-003	Das System soll Protokollabläufe visuell klar darstellen.	
UR-004	Oberflächen sollen weitestgehend mit der Tastatur bedienbar sein.	

4.2 Zuverlässigkeit (Reliability)

4.3 Leistung (Performance)

ID	Beschreibung	Querverweise
PR-001	Bei dem System sollen im Netzwerkmodus keine merkbaren Performanceeinbußen auftreten.	
PR-002	Das System soll direkt auf Nutzereingaben reagieren.	
PR-003	Das System bzw. die Oberfläche soll bei großen Datenmengen reaktionsschnell sein.	
PR-004	Das System soll keine unnötigen Ressourcen anfordern	

4.4 Unterstützbarkeit (Supportability)

ID	Beschreibung	Querverweise
SR-001	Die Oberflächentechnologie soll leicht ersetzbar sein.	
SR-002	Die zentrale Anwendungslogik soll automatisch testbar sein.	
SR-003	Das System verwendet als Standardsprache Deutsch. Zusätzliche Sprachpakete sollen leicht eingebracht werden können.	
SR-004	Der Quellcode soll übersichtlich und leicht wartbar sein.	

4.5 Sonstige Einschränkungen

4.5.1 Schnittstellen

Als Schnittstelle zur Oberfläche steht das .NET Framework in der Version 4.7.03190 zur Verfügung.

4.5.2 Implementierung

Das Projekt muss mit Visual Studio Enterprise 2019 in der Version 16.6.1 lauffähig und analysierbar sein.

5 Risikoakzeptanz

6 Skizze der Gesamtsystemarchitektur

Eine vollständige Systemarchitektur befindet sich im beiliegenden Dokument „Systementwurf.pdf“.

7 Lieferumfang

Die folgende Tabelle enthält alle Arbeitsergebnisse, die in der Veranstaltung „Software-Projekte“ zu dem vom Team zu liefernden „End-Produkt“ gehören – für die individuell von jedem Projektteilnehmer zu liefernden Ergebnisse lesen Sie bitte im Projektleitfaden bzw. im Projektkalender nach. Die Benotung erfolgt nicht nur auf Grundlage des lauffähigen Programms, sondern bezieht die Qualität der Analyse, des Entwurfs und des Systemtests mit ein.

Lfd. Nr.	Was?	Art des Dokuments	Bemerkungen
Ergebnis der System-Analyse			
1	Systemanalyse.pdf	Ausgefüllte Vorlage mit fertiger Systemanalyse	
2	Sequenzdiagramme.pdf		Eigenes Dokument zur besseren Übersicht und Lesbarkeit beigelegt
3	UC-Digramm.pdf		Eigenes Dokument zur besseren Übersicht und Lesbarkeit beigelegt
4	Konzeptuelles_Datenmodell.pdf		Eigenes Dokument zur besseren Übersicht und Lesbarkeit beigelegt
Dokumentation des Systementwurfs			
5	Systementwurf.pdf	Ausgefüllte Vorlage mit fertigem Systementwurf	
6	SoftwareProjekt_Mockup.pdf		Eigenes Dokument zur besseren Übersicht und Lesbarkeit beigelegt
7	SoftwareProjekt_Mockup_AdobeXd.pdf		Eigenes Dokument zur besseren Übersicht und Lesbarkeit beigelegt
Implementierung			
8	Lauffähiger und getesteter Quellcode		Abgabe am Semesterende
Test			
9	Testspezifikation Systemtest		Endgültige Abgabe am Semesterende; zur Vorbereitung des Abnahmetests ist die Aufstellung der in den Abnahmetest einbezogenen Testfälle früher vorzulegen (Termin im Projektkalender)
10	Testprotokoll Systemtest		Abgabe am Semesterende

8 Abnahmekriterien

In der Veranstaltung „Software-Projekte“ werden vom „Auftraggeber“ (in Absprache mit den Teilnehmern) rechtzeitig vor Semesterende Systemtestfälle ausgewählt, die das System dann am Tag der Abnahme ohne Beanstandung „überstehen“ muss.

9 Glossar

Begriff	Erklärung
Alice	Ist ein Teilnehmer des Protokolls. Sendet die Nachrichten.
BB-84-Protokoll	Protokoll zum Austausch eines Schlüssels, welches sich spezielle Eigenschaften der Quantenphysik zunutze macht.
Bob	Ist ein Teilnehmer des Protokolls. Empfängt die Nachrichten.
Cipher-Nachricht	Mit Final-Key verschlüsselte Nachricht, die Alice an Bob sendet.
Datenbit / Bit	Pseudo Bit, im Programm, das ein reales Bit simuliert. Kann die Zustände “0” und “1” annehmen.
Eve	Ist ein Teilnehmer des Protokolls. Versucht die Kommunikation zwischen Alice und Bob abzuhören.
FinalKey	Wird zum Ver-/ Entschlüsseln der Nachricht verwendet.
Lokaler Modus	Modus, bei dem alle Benutzer einen Rechner verwenden, um an der Simulation teilzunehmen.
Netzwerk-Modus	Modus, bei dem jeder Benutzer einen eigenen Rechner verwendet, um an der Simulation teilzunehmen.
Notebook	Datenspeicher, in dem jede Rolle ihre Informationen zu Datenbits, Polarisationen und Keys speichert.
Öffentlicher Kanal	Unverschlüsselter elektronischer Datenkanal. Ist für alle Teilnehmer einsehbar.
Photonen	Teilchen, das über den Quantenkanal versendet wird. Entsteht aus einem Datenbit, das mittels einer Polarisation polarisiert wurde.
Polarisation	Gewählte Richtung des Polarisationsfilters beim Erzeugen und Messen der Photonen.
PreKey	Vorläufiger Schlüssel, der aus den Datenbits gebildet wird, bei denen Alice und Bob übereinstimmende Polarisationen gewählt haben.
Quantenkanal	Kanal, der Photonen zwischen Alice und Bob transportiert. Eve kann diesen kompromittieren.
Rollen	Teilnehmer am Protokoll (Alice, Bob, Eve)
Simulation	Bestandteil des Programms, das den Ablauf des BB-84-Protokolls durchläuft.
System	Umschreibung für das laufende Programm.

10 Abkürzungsverzeichnis

Abkürzung	Erklärung
ASCII	American Standard Code II
IDE	Integrated Development Environment
IP	Internet Protocol
JDK	Java Development Kit
OTH	Ostbayrische Technischen Hochschulen
TCP	Transmission Control Protocol
UC	Use-Case
QCC	QuantenCryptoCram

11 Literaturverzeichnis

[Lar] Larman Craig, *Applying UML And Patterns. An Introduction to Object-Oriented Analysis And Design,* Prentice Hall, 2nd ed., 2002

12 Abbildungsverzeichnis

Abbildung 1: Konzeptionelles Datenmodell 25
Abbildung 2: Objektdiagramm Lokales Spiel (links: mit Eve; rechts: ohne Eve) 26
Abbildung 3: Objektdiagramm Remote Spiel 26