**UC-A-41 Enhanced Login with Indemnity Status**

|  |  |
| --- | --- |
| **Field** | **Value** |
| **Use Case Name** | UC-A-41 Enhanced Login with Indemnity Status |
| **Last Updated** | June 24, 2025 |
| **Status** | Not Started |
| **Assigned** | Augustine Nwagboso |
| **Scenario** | When a user logs in, include indemnity status in JWT token |
| **Post Conditions** | JWT token contains IndemnityAccepted flag |
| **Endpoint Name** | POST /loginv1 (modified) |
| **Controller Method** | LoginV1 (enhanced) |
| **Authorized Roles** | All |
| **Query Parameters** | Existing login parameters |
| **Request Body** | Existing login request body |
| **Response** | json { "statusCode": "string", "status": "string", "message": "string", "data": { "token": "string (includes IndemnityAccepted claim)", "user": { "indemnityAccepted": "boolean" } } } |
| **Flow of Events** | User submits login credentials BEGIN SYSTEM if (!credentials.Valid) { return Error-A-XX (Invalid Login Credentials) } let user = db.Users.Authenticate(credentials) if (user is NULL) { return Error-A-YY (Authentication Failed) } else { let indemnityAcceptedStatus = user.IndemnityAccepted let token = GenerateJwtToken(user, { IndemnityAccepted: indemnityAcceptedStatus }) return { token: token, user: { indemnityAccepted: indemnityAcceptedStatus } } } END SYSTEM User receives JWT token with indemnity status |
| **Required Dependencies** | Account microservice JWT token generation Users database table |
| **Comments** |  |
| **Test Checklist** |  |